

Facultad de Informática y Telecomunicaciones  
Departamento de Telecomunicaciones



**TÍTULO:**

**SISTEMA DE ADQUISICIÓN DE DATOS PARA BATERÍAS DE CASAS DE  
CURAR TABACO**

**TESIS PRESENTADA EN OPCIÓN AL TÍTULO ACADÉMICO DE MÁSTER EN  
SISTEMAS DE TELECOMUNICACIONES**

**Autor:** Ing. José Ramón Hernández Márquez

**Tutor:** Dr. Pedro Antonio Sosa Fernández

**Asesor:** Ing. Frank Ernesto Ramírez Torres

**2009**

## *Agradecimientos*

Como reconocimiento a la valiosa colaboración prestada en la realización de este trabajo, ofrezco mis más sinceros agradecimientos:

En primer lugar a mi esposa e hijos por su comprensión e incondicional apoyo sobre todo en los momentos más difíciles en el transcurso de la maestría.

A toda mi familia que en todo momento se ha preocupado por el éxito de mi superación como profesional.

A mi tutor que siempre estuvo dispuesto a brindarme su conocimiento, tanto en la confección como en la revisión de la tesis.

A mi asesor que sin su valiosa ayuda hubiera sido muy difícil alcanzar los objetivos propuestos en este trabajo dado su gran conocimiento del tema.

A todo el claustro de profesores que se esforzaron por transmitirnos sus conocimientos y habilidades en todas las asignaturas.

A mis compañeros de trabajo por su solidaridad durante todo el tiempo que duró la maestría.

A la Dirección de LACETEL, por su colaboración en el desarrollo y culminación de este trabajo y por propiciar nuestra participación en el estudio de esta carrera.

A todos muchas gracias.

## *Dedicatoria*

Sería interminable mencionar a todas aquellas personas que de una forma u otra han tenido incidencia en que mis estudios de post grado terminen con éxito, por eso:

Este trabajo va dedicado especialmente:

A mi esposa e hijos, que sienten como suyos el éxito alcanzado en este trabajo.

A mis hermanos, por su apoyo en la realización de este trabajo.

A la Revolución que ha creado un sistema de educación que nos da oportunidades a todos por igual.

A Fidel, máximo impulsor de la Revolución Científico Técnica en Cuba.

## *Opinión del Tutor*

## *Nota de Aceptación*

---

---

---

---

---

---

---

---

---

---

---

---

Presidente del Tribunal

---

Secretario

---

Vocal

Ciudad de Pinar del Río

Fecha \_\_\_\_\_

Cuño

| Índice         |  | Página    |
|----------------|--|-----------|
|                | <b>Resumen</b>   |           |
|                | <b>Introducción</b>  | <b>1</b>  |
|                | <b>Capítulo 1: Descripción y evolución de las tecnologías aplicadas a la cura del tabaco</b> | <b>4</b>  |
| <b>1</b>       | <b>Tecnologías aplicadas en el proceso de cura controlada del tabaco</b>                     | <b>4</b>  |
| <b>1.1</b>     | <b>Estado actual de la temática</b>  | <b>4</b>  |
|                | <b>Capítulo 2: Descripción del Sistema de Adquisición de Datos</b>                           | <b>10</b> |
| <b>2.1</b>     | <b>Diseño del Sistema de Adquisición de datos</b>  | <b>10</b> |
| <b>2.2</b>     | <b>Sistema de adquisición de Datos de la casa de curado</b>                                  | <b>12</b> |
| <b>2.3</b>     | <b>Lector</b>  | <b>12</b> |
| <b>2.4</b>     | <b>Programa residente en la PC</b>   | <b>13</b> |
| <b>2.5</b>     | <b>Características técnicas del PIC 16LF73</b>   | <b>13</b> |
| <b>2.5.1</b>   | <b>Memoria de programa</b>   | <b>15</b> |
| <b>2.5.2</b>   | <b>El contador de programas</b>  | <b>16</b> |
| <b>2.5.3</b>   | <b>Memoria de datos</b>  | <b>17</b> |
| <b>2.5.3.1</b> | <b>Registros de propósito general (GPR)</b>  | <b>18</b> |
| <b>2.5.3.2</b> | <b>Registros de funciones especiales (SFR)</b>   | <b>19</b> |
| <b>2.5.4</b>   | <b>Interrupciones</b>  | <b>23</b> |
| <b>2.5.5</b>   | <b>Puertos de entrada – salida</b>   | <b>23</b> |
| <b>2.5.6</b>   | <b>Set de instrucciones</b>  | <b>24</b> |
| <b>2.6</b>     | <b>Características técnicas del MRF24J40 Transmisor/Receptor</b>                             | <b>28</b> |
| <b>2.6.1</b>   | <b>Interrupciones del MRF24J40</b>   | <b>30</b> |

|                |  |           |
|----------------|--|-----------|
| <b>2.6.2</b>   | <b>Estructura de la Interrupción</b>   | <b>31</b> |
| <b>2.6.3</b>   | <b>Estructura del paquete IEEE 802.15.4</b>                                  | <b>33</b> |
| <b>2.6.4</b>   | <b>Formato FIFO de transmisión</b>   | <b>36</b> |
| <b>2.6.5</b>   | <b>Recepción de paquetes</b>   | <b>38</b> |
| <b>2.6.6</b>   | <b>Componentes periféricos del MRF24J40</b>                                  | <b>41</b> |
| <b>2.6.7</b>   | <b>Organización de la memoria del MRF24J40</b>                               | <b>42</b> |
| <b>2.6.8</b>   | <b>Tarjeta RF del MRF24J40 2.46 GHZ</b>                                      | <b>49</b> |
| <b>2.6.8.1</b> | <b>Características generales de la antena</b>                                | <b>51</b> |
|                | <b>Capítulo 3 Aspectos prácticos del proyecto</b>                            | <b>55</b> |
| <b>3.1</b>     | <b>Diseño y construcción del Sistema de Adquisición de Datos</b>             | <b>55</b> |
| <b>3.2</b>     | <b>Soluciones de software para el SAD</b>                                    | <b>57</b> |
| <b>3.3</b>     | <b>Diseño y construcción del lector</b>                                      | <b>64</b> |
| <b>3.4</b>     | <b>Soluciones de software para el lector</b>                                 | <b>67</b> |
| <b>3.5</b>     | <b>Herramientas de diseño utilizadas durante la elaboración del proyecto</b> | <b>70</b> |
| <b>3.6</b>     | <b>Programa residente en la PC para a descarga de los datos</b>              | <b>71</b> |
| <b>3.7</b>     | <b>Gastos del proyecto</b>   | <b>76</b> |
|                | <b>Conclusiones</b>  | <b>80</b> |
|                | <b>Bibliografía</b>  | <b>82</b> |
|                | <b>Referencias bibliográficas</b>  | <b>83</b> |

## *Resumen*

En el presente trabajo de diploma se hace una descripción de un sistema de adquisición de datos diseñado y construido a nivel de maqueta con el objetivo de ser aplicado en el proceso de curado del tabaco en secaderos artificiales conocidos por KALFRISAS.

Este trabajo constituye la segunda fase de un proyecto que se desarrolla a petición del Instituto de Investigaciones del Tabaco y tiene como objetivo obtener los datos necesarios durante el proceso de curado de la hoja con el fin de poder evaluar la calidad del proceso y la eficiencia y eficacia con que éste transcurrió.

Se hace un análisis del estado actual del arte sobre esta temática tanto en nuestro país como fuera de éste y una descripción completa del sistema de adquisición de datos, que consta de dos elementos: el SAD y el lector, incluyendo el hardware y el software de cada uno de los dos elementos que lo conforman.



## *Introducción:*

En la actualidad en el mundo se está desarrollando cada vez más la industrialización de la agricultura y una de las ramas de ésta en la que se han logrado avances científicos de gran significación es en el proceso de curar el tabaco. Estos adelantos de la ciencia se aplican con el objetivo de lograr una calidad cada día mayor de los productos exportables derivados de la hoja de esta planta. Existen firmas en el mundo como la KALFRISA y otras que han desarrollado una serie de equipos que se emplean en la cura de la preciada hoja. La UCT LACETEL ha desarrollado controladores de humedad y temperatura que cumplen con los requisitos internacionales para el curado del tabaco, llegando incluso a contar con un alto grado de integración y de inteligencia asociada a la aplicación, como son rampas de temperatura que hacen más eficiente esta actividad. En la actualidad se hace cada vez más necesario el ahorro de portadores energéticos y una forma de lograrlo es haciendo cada vez más eficiente el proceso de secado de la hoja, ya que los quemadores de las casas de curado son grandes consumidores de combustible diesel, siendo éste uno de los parámetros que controla y chequea el sistema que se propone desarrollar.

En las Empresas Tabacaleras del país se realiza la cura de algunas variedades de tabaco en casas especiales destinadas a este fin, conocidas por KALFRISAS, ya que su tecnología proviene de esta firma comercial. En estas casas se controlan parámetros tales como la humedad y la temperatura durante todo el proceso, lo que se garantiza por medio de controladores que gobiernan el encendido del quemador y la apertura de ventanillos presentes en dicha instalación. Para el control de este proceso se gastan grandes sumas de dinero importando equipos que en muchos casos no responden a las necesidades reales propias de nuestro país. Uno de los parámetros que actualmente es importante monitorear es el consumo del combustible utilizado en el proceso de cura, es por eso que el Instituto del Tabaco encargó a la UCT LACETEL de Pinar del Río el desarrollo de un sistema que fuera capaz de monitorear los parámetros

fundamentales del curado del tabaco y además brindara la información de la cantidad de combustible consumido durante el tiempo que dura este proceso, sin que esta información pueda ser conocida por el operador y mucho menos adulterada. La tecnología con que se dispone en el país para el curado del tabaco no es capaz de almacenar la información sobre el proceso, ni tampoco de brindar datos de importancia como son el tiempo y la cantidad de combustible empleados durante una corrida, lo que constituye el **problema a resolver** en este trabajo. Se necesita pues un sistema que dé seguimiento a estos parámetros con el objetivo de tomar decisiones oportunas sobre el proceso.

**El objeto** de esta investigación lo constituye la búsqueda de soluciones para la medición y almacenamiento de los datos del proceso tecnológico de curado del tabaco, mientras que el **campo de acción** lo constituye el estudio de la transmisión y recepción de los datos adquiridos con la requerida calidad en las condiciones en que se realiza la actividad que se quiere monitorear.

Para la elaboración de este trabajo se tiene como **objetivo general**:

Diseñar un sistema de adquisición de datos (SAD) que sea capaz de almacenar y transmitir de forma inalámbrica a un dispositivo electrónico (lector) todos los datos inherentes al proceso de curado del tabaco que se efectúa en una batería de casas utilizadas con este fin, con el objetivo de evaluar la calidad y la eficiencia del proceso.

Y como objetivos específicos:

1. Diseñar y evaluar a nivel de maqueta un conjunto de soluciones técnicas de hardware y software para registrar los datos adquiridos en el sistema (parámetros tecnológicos solicitados) y establecer la comunicación inalámbrica entre el SAD y el lector, utilizando el microcontrolador PIC16LF73 y el chip MRF24J40 para la comunicación inalámbrica.
2. Utilizar el protocolo de comunicación IEEE 802.15.4, de forma tal que permita la transmisión de datos entre el microcontrolador y el MRF24J40

por medio del puerto SPI y entre el SAD y el lector en ambas direcciones de forma inalámbrica.

3. Asimilar y utilizar el chip de comunicaciones MRF24J40 con el propósito de utilizarlo en futuros proyectos de automatización.

**La hipótesis** a defender será la siguiente:

Es posible diseñar un sistema de comunicación bidireccional inalámbrica que sea capaz de almacenar y transmitir los datos del proceso de curado del tabaco durante una corrida de éste, y que éstos puedan ser llevados y procesados en una computadora.

## CAPÍTULO 1: Descripción y evolución de las tecnologías aplicadas a la cura del tabaco.

### 1. Tecnologías aplicadas en el proceso de la cura controlada del tabaco.

#### 1.1 Estado actual de la temática.

Los sistemas tecnológicos más generalizados en el mundo son los denominados CEPA y MaxiMiser [1] que se describen brevemente a continuación.

- a) Secado Automático de Tabaco Virginia - Sistema cepa (Control de Estufado de Productos Agrícolas). Se trata de un sistema electrónico, que consta de una central de Control capaz de manejar hasta ocho (8) hornos de secado a través de periféricos (Fig. 1.1).

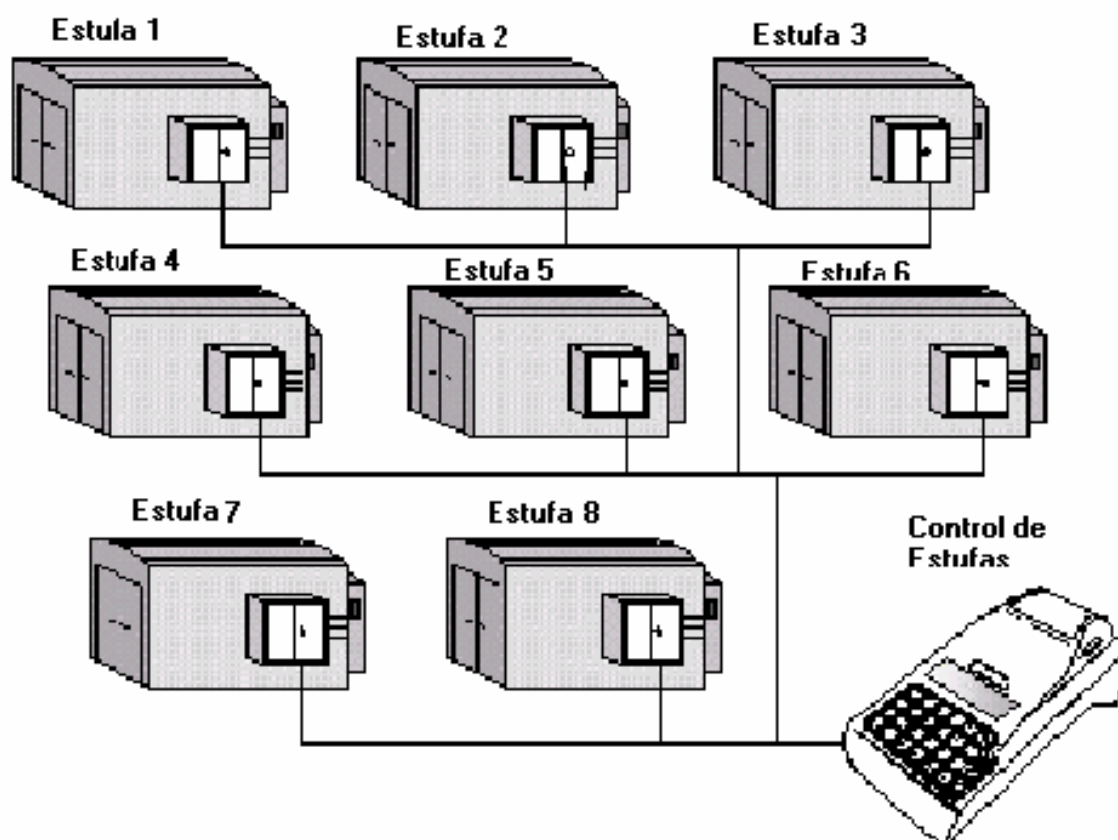


Figura 1.1 Sistema CEPA, control de estufado de productos agrícolas.

Esta versión del producto está expresamente realizada para el Tabaco Blanco Virginia, producido principalmente en las provincias de Salta y Jujuy, en Argentina. La Central maneja las diferentes fases de curado y secado permitiendo la realización de estufadas conformes con los valores recomendados (“curvas”, “tablas”, parámetros”). Además genera reportes y estadísticas del ciclo.

Los periféricos están diseñados para adaptarse virtualmente a cualquier tipo de estufa Bulk Curing, controlando los principales elementos: quemador, ventana (a través de un motor), sensores de temperatura y de humedad y alarmas.

El sistema fue desarrollado entre la Empresa Soluciones P.C.P.S.A., y CITE S.P.A. (Italia).

Las pruebas realizadas en los últimos años, certifican al sistema “CEPA” como apto y de suma utilidad, porque:

- Reduce el período de secado sin perjudicar la calidad de la hoja.
- Mejora la calidad del producto (hojas más uniformes, estabilidad de temperatura y humedad durante el ciclo).
- Aumenta la cantidad de tabaco curado (posibilidad de “sobrecargar” la estufa).
- Reduce el consumo de combustible y permite detectar fallas de las estufas.
- Elimina los riesgos: deshidratación, rotura de las hojas e incendios.

b) MaxiMiser Bulk Curing Systems. [1] Es el modelo típico de hornos para la cura de tabaco Virginia, normalmente conocidos como “Bulk Curing” (Fig. 1.2). Existen dos modelos de hornos, que se diferencian en la cantidad de hojas a curar. Ambos están contruidos con estructuras de acero y poseen excelentes características de hermeticidad.



Figura 1.2 Horno Bulk Curing

Éstos se comercializan como todo un sistema, el cual tiene incluido los dispositivos utilizados para el control del proceso.

Entre sus principales características destacan las siguientes:

- Flujo de aire ajustable: se puede regular el aire que pasa entre las hojas durante las diferentes etapas de la cura, en la salida del horno.
- El quemador entrega todo el calor necesario para una cura eficiente.
- La construcción de la cámara de combustión con acero inoxidable garantiza una larga vida al sistema.
- Gran capacidad de cura, dada por el uso de ventiladores de motor jaula de ardilla que permiten mover grandes cantidades de aire y mantener con eficiencia el ambiente en el interior del secadero.
- Hornos con paredes aisladas térmicamente para incrementar eficiencia.
- Protección del horno contra fallas del flujo de aire.
- Protección por sobre temperatura.
- Control de temperatura Automático.
- Control manual de la Humedad Relativa.

Los ejemplos anteriores sirven como referencia de la tecnología que se utiliza en el mundo para la actividad, pero en el contexto cubano, son otros sistemas los que se emplean:

- El SCOPETTI.
- El Kalfrisa.

El sistema SCOPETTI es de fabricación Italiana, mientras que el Kalfrisa es de procedencia Española, siendo éste el más utilizado. En el caso de Kalfrisa existen dos modelos de generadores que se diferencian en el caudal de aire, 32000 y 20000 m<sup>3</sup>/h respectivamente, con una capacidad calorífica de 200000 Kcal/h, una potencia instalada por equipo máxima de 10 Kw/h, partiendo de dos ventiladores centrífugos accionados por motores eléctricos y un quemador diesel (Fig.1.3).



Figura 1.3 Generadores Kalfrisa

Además, como antecedente al trabajo que se presenta en esta Tesis se tiene el contador de tiempo real de trabajo para motores de regadío de combustión interna y kalfrisas de curar tabaco, desarrollado en la UCT de Base LACETEL Pinar del Río.

A continuación se describe de forma abreviada este trabajo.

El control de tiempo para motores de regadío es un equipo que es capaz de registrar el tiempo de trabajo de un motor de regadío de combustión interna, o del quemador de una casa de curado, sin que se pueda alterar su funcionamiento, ni acceder a la información por él registrada, por personal alguno que no haya sido previamente autorizado e instruido a tal efecto.

El equipo diseñado consta de tres elementos:

- Contador de tiempo real
- Lector portátil
- Software de comunicación

### **Descripción del contador**

El contador está diseñado a base de un microcontrolador de la familia PIC16LF73 de tecnología MOS. En el diseño se usó un cristal de cuarzo de 32,768 kHz que proporciona la base de tiempo necesaria para garantizar la exactitud en la medición del tiempo y permite además reducir el consumo del equipo al orden de los microamperes a plena capacidad de uso.

La forma de sensado puede ser ajustada a las condiciones que requiera la instalación a la que se le quiera medir el tiempo real de trabajo.

Este dispositivo no posee ningún órgano de mando o control, por lo que no se puede interactuar con él si no es por medio del lector, de esta manera no se puede conocer el tiempo de trabajo registrado por éste hasta que los datos no sean descargados en la PC.

Se alimenta con dos baterías de 1,5V del tipo AA. Tiempo de duración mayor de 12 meses.

### **Descripción del lector portátil**

Al igual que el contador, el lector está montado en un microcontrolador PIC 16LF73, lo que en primera instancia garantiza su portabilidad, debido a las pequeñas dimensiones del impreso que se logra con su utilización y al bajo consumo de energía de este microcontrolador.



Como su nombre lo indica, el lector es el encargado de recolectar los datos almacenados en la memoria del contador. Éste es capaz de leer los registros almacenados por varios equipos, a los que en el momento de la instalación se les asignará un número de identificación. La comunicación entre ambos se realiza por medio de luz infrarroja. Este método de comunicación se emplea para garantizar que las difíciles condiciones de explotación, humedad, polvo en el ambiente, grasa, etc. no puedan influir y por tanto distorsionar o impedir la transmisión hacia el lector portátil de los datos almacenados en la memoria del contador.

El lector portátil, al igual que el contador, no posee ningún órgano que visualice los datos almacenados en él, por lo que el personal encargado de hacer esta operación no puede en ese momento conocer el tiempo de trabajado del motor, o los motores a los que le realice la lectura.

Los datos que se encuentran en la memoria del lector son descargados en una PC a la que previamente se le ha instalado un software producido a tal efecto por el equipo de diseño.

La comunicación de la PC con el lector se realiza por puerto serie RS-232

### **Características técnicas del lector portátil**

Se alimenta con cuatro baterías de 1,5V del tipo AA, tiempo de duración mayor de 12 meses. Con el objetivo de prolongar el tiempo de duración de las mismas cuando no se realiza ninguna operación por un tiempo mayor de dos minutos y medio el equipo se autodesconecta.

## CAPÍTULO 2: Descripción del Sistema de Adquisición de Datos

### 2.1. Diseño del Sistema de Adquisición de Datos

El trabajo desarrollado en la tesis es parte de un proyecto que consiste en diseñar un sistema de adquisición de datos (SAD) para el proceso de curado del tabaco en el que se obtengan las principales variables que intervienen en esta actividad, como son: la temperatura, la humedad relativa, el consumo de combustible y el tiempo de encendido del quemador de las casas de curado. En la figura 2.1 se muestra un diagrama que permite tener una idea del alcance del proyecto que da origen a esta tesis.

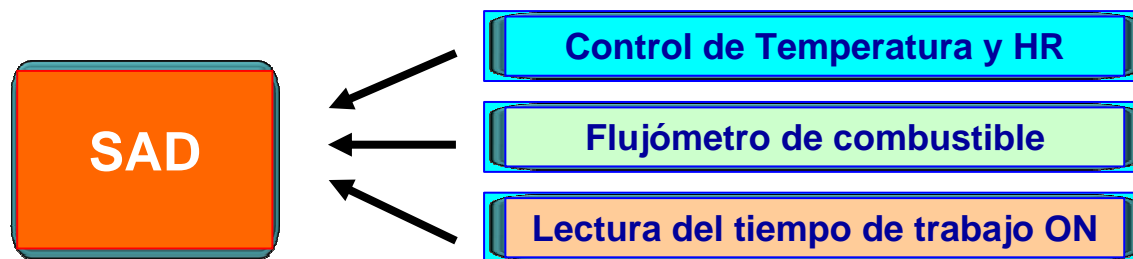


Figura 2.1. SAD

El sistema de adquisición de datos, objeto fundamental de este trabajo, consiste en la modernización del contador de tiempo real de trabajo para motores de regadío de combustión interna y kalfrisas de curar tabaco que fue descrito en el capítulo anterior.

Este sistema en su conjunto (ver figura 2.2) estará compuesto por Varios SADs (pueden ser hasta doce) que se ubicarán en las kalfrisas y que serán los encargados de almacenar los datos fundamentales del proceso de cura del tabaco, como son: temperatura y humedad relativa, combustible consumido y tiempo de duración del proceso. Estos datos brindan la posibilidad al personal técnico de poder evaluar la eficiencia del proceso de cura del tabaco desde que éste inicia hasta que termina.

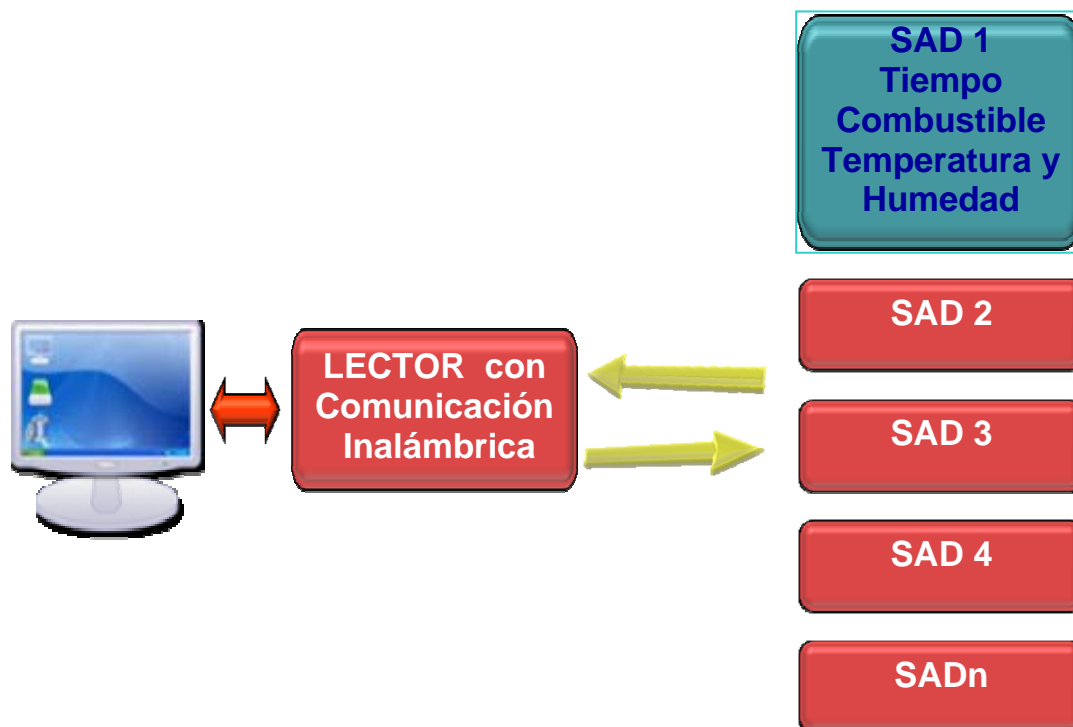


Figura 2.2. Diagrama del sistema completo del SAD.

Después de haber hecho un análisis de las tecnologías con que se dispone para resolver el problema planteado en esta tesis, se decidió diseñar un Sistema de adquisición de Datos (SAD) donde se utiliza el chip transceptor MRF24J40 conectado a un microprocesador PIC16LF73 mediante su interfaz SPI, ya que esto permite la comunicación inalámbrica en ambos sentidos con todos los SADs instalados en cada casa de curar tabaco a una distancia de aproximadamente 60 metros.

El MRF24J40 se controla desde el PIC para realizar las funciones de las capas PHY y MAC, permitiendo modular QPSK y la TX inalámbrica, de esta forma se logra la comunicación entre el SAD y un lector, el cual será el encargado de trasladar la información a una PC para su posterior análisis.

Como puede observarse en la figura 2.2, el sistema está compuesto por:

SAD (pueden ser hasta 12 sistemas).

Lector

Programa residente en la PC

A continuación se hará un análisis detallado de cada uno de estos elementos que componen el sistema.

## **2.2. Sistema de Adquisición de Datos de la casa de curado.**

El sistema de adquisición de datos es el encargado de recibir los datos del proceso de curado del tabaco. El esquema eléctrico se sustenta en un microprocesador de la familia PIC, el PIC16LF73, y el Chip tranceptor MRF24J40, las características técnicas de ambos se describen más adelante en este mismo capítulo.

### **Funciones a realizar por el SAD**

1. Recibe y almacena los datos de humedad y temperatura del control de temperatura y humedad que tendrá una interfase con el microcontrolador del SAD para la transmisión de los datos.
2. Recibe y almacena en memoria los datos sobre el consumo de combustible provenientes del flujómetro instalado en la entrada de combustible del quemador de la kalfrisa.
3. Cuenta el tiempo de encendido de la kalfrisa.
4. Transmite todos los datos cuando es encuestado por el lector.

## **2.3. Lector**

El lector es el encargado de recolectar los datos de todos los SADs y transferirlos a la PC por medio de un programa residente en ésta, los que son utilizados por los especialistas para tomar decisiones en el proceso de cura del tabaco.

## **2.4. Programa residente en la PC**

La información almacenada en la memoria del lector será descargada a una PC vía RS-232, para lo cual se contará con un programa desarrollado en LabWindows/CVI que permitirá almacenar los datos de todas las casas de curado para su posterior procesamiento. Al igual que ocurre con el SAD, los datos almacenados en la memoria del lector no podrán ser borrados hasta que no se haya efectuado la operación de lectura.

## **2.5. Características técnicas del PIC16LF73.**

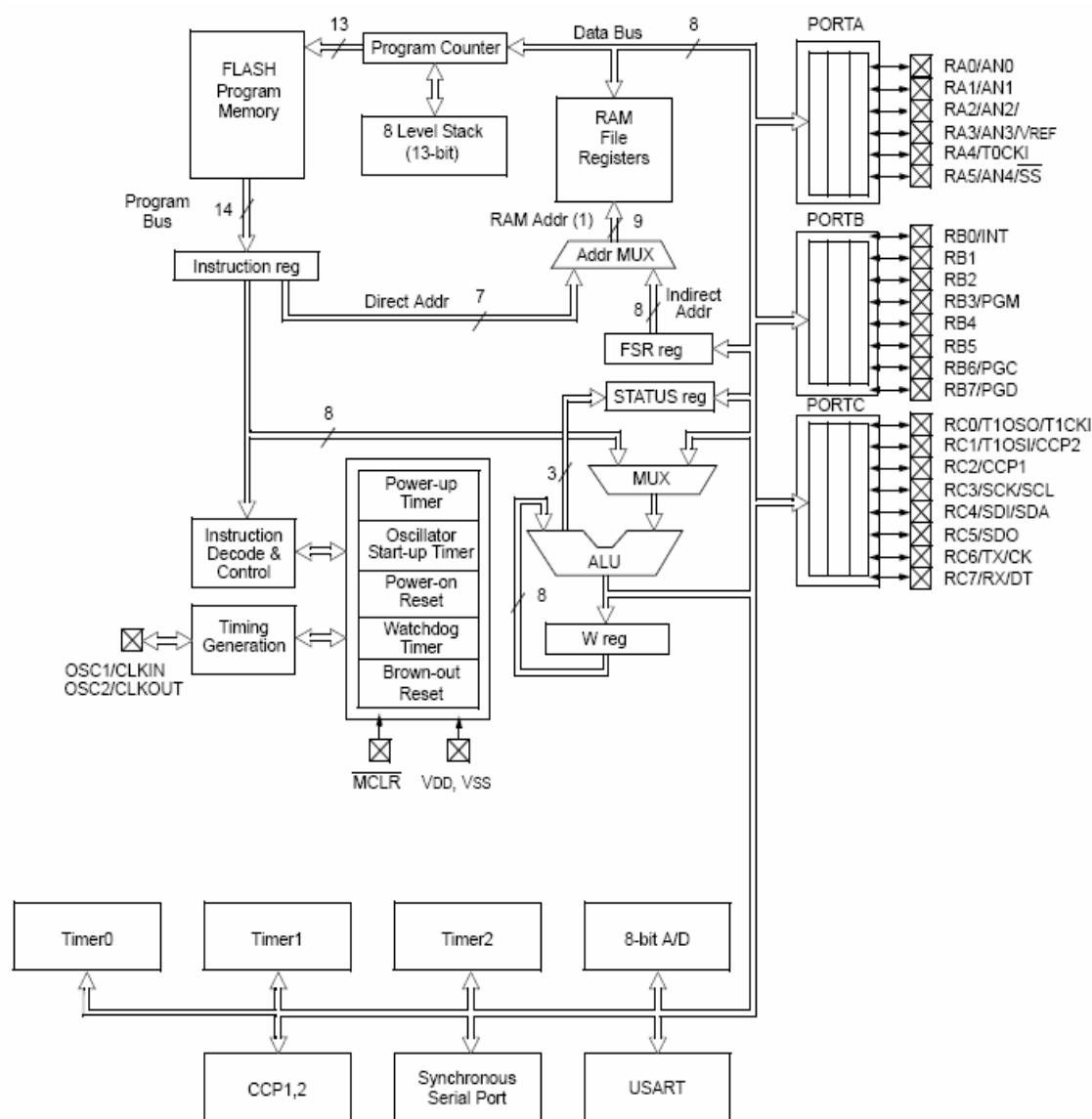
Tanto el sistema de adquisición de datos como el lector que conforman el sistema a desarrollar están montados sobre el microcontrolador PIC16LF73. La familia LF está concebida para trabajar con niveles de voltajes tan bajos como 2V y una pequeña corriente de consumo, típicamente de 2mA trabajando con 5V y cristal de 4 MHz y 20 uA alimentado con 3V y cristal de 32 KHZ, lo que lo hace apropiado para ser usado en sistemas alimentados con baterías.

Como todos los microcontroladores actuales, éste está diseñado según la arquitectura Harvard, por lo que dispone de dos memorias independientes: una que está disponible sólo para instrucciones y otra solo para datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias, todo esto propicia que se mejore la velocidad de trabajo con respecto a la arquitectura tradicional (Von Neumann).

Este microcontrolador tiene un conjunto de instrucciones, que hace posible que se lleve a cabo cualquier operación en cualquier registro usando cualquier modo de dirección, lo cual hace la programación con este microcontrolador simple y más eficiente.

Los microcontroladores PIC16LF73 [2] poseen una Unidad Aritmética Lógica (ALU) y un registro de funciones de 8 bits (ver figura 2.3). Ésta realiza funciones aritméticas tales como: adición, sustracción, cambio o intercambio de nibbles o bytes entre registros, operaciones lógicas y funciones entre los datos, en el registro de trabajo y en cualquier archivo de registro. En dependencia de la

instrucción ejecutada, la ALU del procesador puede afectar los valores (imponiendo el valor del 1 lógico ó el valor del 0 lógico), lo que corresponde al seteo o al reseteo de cada uno de los bits del registro de estado, que son: el acarreo o Carry (C), el acarreo digital o Digit Carry (la CD), y el cero o Zero (Z) bits en el registro de estado.



Note 1: Higher order bits are from the STATUS register.

Figura 2.3 Diagrama en bloques del PIC16LF73

En la tabla 1 se muestran las características más importantes del microprocesador PIC16LF73 y en la figura 2.4 se muestra su diagrama de pines.

Tabla 1. Principales características del PIC 16LF73

|                                   |                       |
|-----------------------------------|-----------------------|
| Frecuencia de operación           | DC- 20MHZ             |
| Resets y demoras                  | POR, BOR, (PWRT, OST) |
| Memoria flash de programa         | 4K                    |
| Memoria de datos                  | 192                   |
| Interrupciones                    | 11                    |
| Puertos de entrada - salida       | Puertos A,B,C         |
| Temporizadores                    | 3                     |
| Comunicación serie                | SSP, USART            |
| Módulo analógico digital de 8-bit | 5 canales de entrada  |
| Set de instrucciones              | 35 instrucciones      |

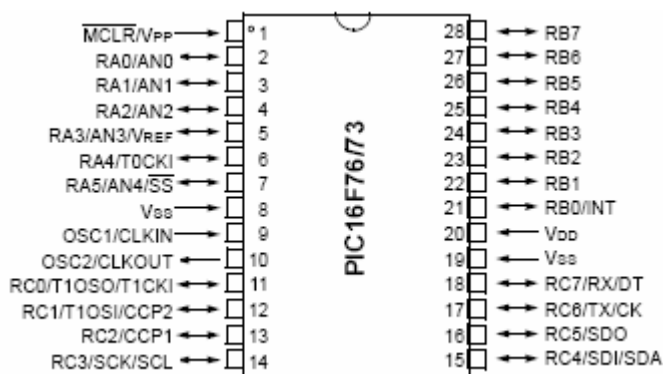


Figura 2.4 Diagrama de pines del PIC16LF73

### 2.5.1. Memoria de programa

El PIC 16LF73 es capaz de direccionar un bloque continuo de memoria de programa de 8K. Las instrucciones CALL y GOTO proveen solo 11 bits de direccionado para permitir la bifurcación dentro de cualquiera de las dos páginas de memoria de 2K (página 0 y página1)

La página 0 comienza en la dirección 0005h y termina en la 07FFh, mientras que la página 1 se extiende desde la dirección 0800h hasta la dirección 0FFFh.

El vector de reset se encuentra en la dirección 0000h, mientras que el vector de interrupción se encuentra en la dirección 0004h. En la figura 2.5 se muestra el mapa de memoria y la pila que está compuesta por ocho niveles.

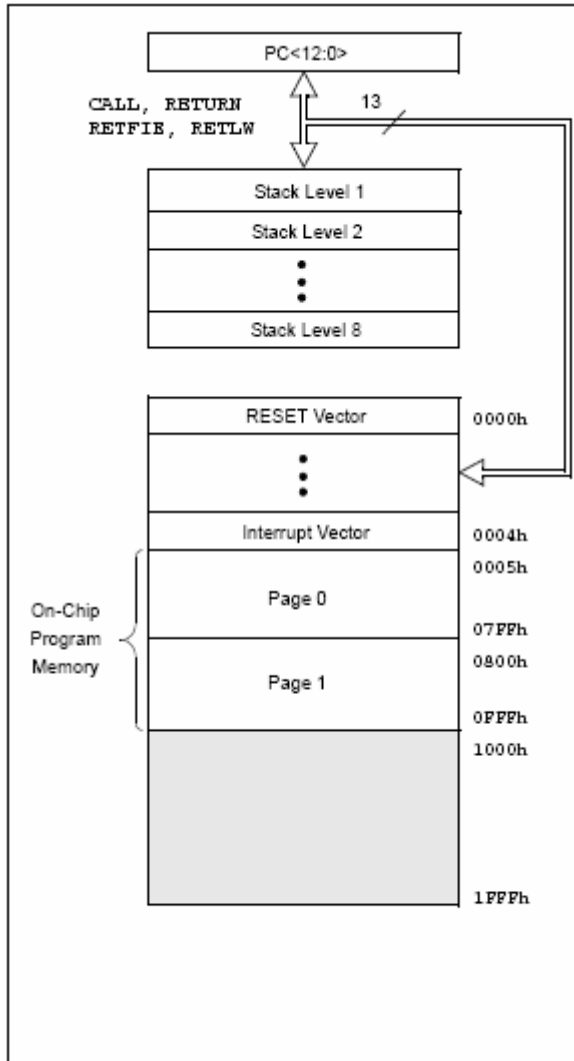


Figura 2.5 Mapa de la memoria de programa y la pila

### 2.5.2. El contador de programa

El microcontrolador PIC16LF73 posee un contador de programa de 13 bits que es el encargado de apuntar a la próxima instrucción a ser buscada para su ejecución. El byte más bajo se encuentra en el registro PCL, el que es leíble y



escribible. El bit más alto (PC<12:8>) no es leíble pero puede ser leído a través del registro PCLATH. Con cualquier reset el bit más alto del contador de programas será limpiado. La figura 2.6 muestra las dos formas que existen de cargar el contador de programas. El ejemplo mostrado en la parte superior de la figura muestra como el contador de programa (PC) es cargado mediante una escritura en el PCL (PCLATH<4:0> → PCH). En la parte inferior de la figura se muestra como el PC es cargado durante una instrucción CALL o GOTO (PCLATH<4:3> → PCH).

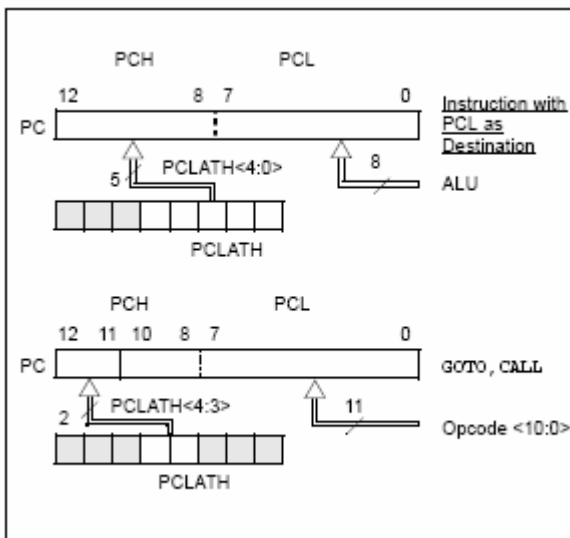


Figura 2.6 Formas de cargar el PC

### 2.5.3 Memoria de datos

La memoria de datos del PIC16LF73 está particionada en cuatro bancos, los que contienen los Registros de Propósito General (GPR) y los Registros Funciones Especiales (SFR).

La selección del banco que se necesite usar se hace a través del registro STATUS mediante los bits RP1 y RP0 como se muestra en la tabla 2.

Tabla2

| RP1:RP0 | Bank |
|---------|------|
| 00      | 0    |
| 01      | 1    |
| 10      | 2    |
| 11      | 3    |

Cada banco se extiende hasta 7Fh (128 bytes). La localización más baja de cada banco está reservada para los Registro de Funciones Especiales. Los Registros de Propósito General están implementados como memoria RAM estática.

Todos los bancos implementados contienen Registros de Funciones Especiales, los más usados en un banco pueden ser reflejados en otro banco, lo que disminuye el código a la hora de programar y se hace más rápido el acceso.

#### **2.5.3.1 Registros de Propósito General (GPR)**

Los registros de propósito general pueden ser accedidos de forma directa o indirectamente a través del Registro de Funciones Especiales (SFR).

Tabla 3. Registros de propósito general y sus direcciones.

| File Address                         | File Address                         | File Address           | File Address           |
|--------------------------------------|--------------------------------------|------------------------|------------------------|
| Indirect addr.(*) 00h                | Indirect addr.(*) 80h                | Indirect addr.(*) 100h | Indirect addr.(*) 180h |
| TMR0 01h                             | OPTION_REG 81h                       | TMR0 101h              | OPTION_REG 181h        |
| PCL 02h                              | PCL 82h                              | PCL 102h               | PCL 182h               |
| STATUS 03h                           | STATUS 83h                           | STATUS 103h            | STATUS 183h            |
| FSR 04h                              | FSR 84h                              | FSR 104h               | FSR 184h               |
| PORTA 05h                            | TRISA 85h                            |                        |                        |
| PORTB 06h                            | TRISB 86h                            | PORTB 105h             | TRISB 185h             |
| PORTC 07h                            | TRISC 87h                            |                        |                        |
| PORTD <sup>(1)</sup> 08h             | TRISD <sup>(1)</sup> 88h             |                        |                        |
| PORTE <sup>(1)</sup> 09h             | TRISE <sup>(1)</sup> 89h             |                        |                        |
| PCLATH 0Ah                           | PCLATH 8Ah                           | PCLATH 10Ah            | PCLATH 18Ah            |
| INTCON 0Bh                           | INTCON 8Bh                           | INTCON 10Bh            | INTCON 18Bh            |
| PIR1 0Ch                             | PIE1 8Ch                             | PMDATA 10Ch            | PMCON1 18Ch            |
| PIR2 0Dh                             | PIE2 8Dh                             | PMADR 10Dh             |                        |
| TMR1L 0Eh                            | PCON 8Eh                             | PMDATH 10Eh            |                        |
| TMR1H 0Fh                            |                                      | PMADRH 10Fh            |                        |
| T1CON 10h                            |                                      |                        |                        |
| TMR2 11h                             |                                      |                        |                        |
| T2CON 12h                            | PR2 92h                              |                        |                        |
| SSPBUF 13h                           | SSPADD 93h                           |                        |                        |
| SSPCON 14h                           | SSPSTAT 94h                          |                        |                        |
| CCPR1L 15h                           |                                      |                        |                        |
| CCPR1H 16h                           |                                      |                        |                        |
| CCP1CON 17h                          |                                      |                        |                        |
| RCSTA 18h                            | TXSTA 98h                            |                        |                        |
| TXREG 19h                            | SPBRG 99h                            |                        |                        |
| RCREG 1Ah                            |                                      |                        |                        |
| CCPR2L 1Bh                           |                                      |                        |                        |
| CCPR2H 1Ch                           |                                      |                        |                        |
| CCP2CON 1Dh                          |                                      |                        |                        |
| ADRES 1Eh                            |                                      |                        |                        |
| ADCON0 1Fh                           | ADCON1 9Fh                           |                        |                        |
| 20h                                  | A0h                                  |                        |                        |
|                                      |                                      |                        |                        |
| General Purpose Register<br>96 Bytes | General Purpose Register<br>96 Bytes | accesses 20h-7Fh       | accesses A0h - FFh     |
| Bank 0 7Fh                           | Bank 1 FFh                           | Bank 2 17Fh            | Bank 3 1FFh            |

■ Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.  
**Note 1:** These registers are not implemented on 28-pin devices.

### 2.5.3.2 Registros de Funciones Especiales (SFR)

Los registros de funciones especiales son registros usados por la CPU y los módulos periféricos para controlar las operaciones del PIC. Estos registros están implementados como memoria RAM estática. La lista de estos registros se muestra en la tabla 4.

Tabla 4. Registros de funciones especiales

| Address              | Name    | Bit 7  | Bit 6   | Bit 5   | Bit 4  | Bit 3           | Bit 2             | Bit 1   | Bit 0   | Value on:<br>POR,<br>BOR | Value on<br>all other<br>RESETS <sup>(2)</sup> |
|----------------------|---------|--|---------|---|--|-----------------|-------------------|---------|---------|--------------------------|--|
| Bank 0               |         |  |         |   |  |                 |                   |         |         |                          |  |
| 00h <sup>(4)</sup>   | INDF    | Addressing this location uses contents of FSR to address data memory (not a physical register) |         |   |  |                 |                   |         |         | 0000 0000                | 0000 0000                                      |
| 01h                  | TMR0    | Timer0 Module's Register   |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 02h <sup>(4)</sup>   | PCL     | Program Counter's (PC) Least Significant Byte  |         |   |  |                 |                   |         |         | 0000 0000                | 0000 0000                                      |
| 03h <sup>(4)</sup>   | STATUS  | IRP  | RP1     | RP0   | $\overline{TO}$  | $\overline{PD}$ | Z                 | DC      | C       | 0001 1xxxx               | 000q quuuu                                     |
| 04h <sup>(4)</sup>   | FSR     | Indirect data memory address pointer   |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 05h                  | PORTA   | —  | —       | PORTA Data Latch when written: PORTA pins when read |  |                 |                   |         |         | --0x 0000                | --0u 0000                                      |
| 06h                  | PORTB   | PORTB Data Latch when written: PORTB pins when read  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 07h                  | PORTC   | PORTC Data Latch when written: PORTC pins when read  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 08h <sup>(5)</sup>   | PORTD   | PORTD Data Latch when written: PORTD pins when read  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 09h <sup>(5)</sup>   | PORTE   | —  | —       | —   | —  | —               | RE2               | RE1     | RE0     | ---- -xxxx               | ---- -uuu                                      |
| 0Ah <sup>(1,4)</sup> | PCLATH  | —  | —       | —   | Write Buffer for the upper 5 bits of the Program Counter |                 |                   |         |         | ---0 0000                | ---0 0000                                      |
| 0Bh <sup>(4)</sup>   | INTCON  | GIE  | PEIE    | TOIE  | INTE   | RBIE            | TOIF              | INTF    | RBIF    | 0000 000x                | 0000 000u                                      |
| 0Ch                  | PIR1    | PSPIF <sup>(3)</sup>   | ADIF    | RCIF  | TXIF   | SSPIF           | CCP1IF            | TMR2IF  | TMR1IF  | 0000 0000                | 0000 0000                                      |
| 0Dh                  | PIR2    | —  | —       | —   | —  | —               | —                 | —       | CCP2IF  | ---- ---0                | ---- ---0                                      |
| 0Eh                  | TMR1L   | Holding register for the Least Significant Byte of the 16-bit TMR1 Register                    |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 0Fh                  | TMR1H   | Holding register for the Most Significant Byte of the 16-bit TMR1 Register                     |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 10h                  | T1CON   | —  | —       | T1CKPS1   | T1CKPS0  | T1OSCEN         | T1SYNC            | TMR1CS  | TMR1ON  | --00 0000                | --uu uuuu                                      |
| 11h                  | TMR2    | Timer2 Module's Register   |         |   |  |                 |                   |         |         | 0000 0000                | 0000 0000                                      |
| 12h                  | T2CON   | —  | TOUTPS3 | TOUTPS2   | TOUTPS1  | TOUTPS0         | TMR2ON            | T2CKPS1 | T2CKPS0 | -000 0000                | -000 0000                                      |
| 13h                  | SSPBUF  | Synchronous Serial Port Receive Buffer/Transmit Register                                       |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 14h                  | SSPCON  | WCOL   | SSPOV   | SSPEN   | CKP  | SSPM3           | SSPM2             | SSPM1   | SSPM0   | 0000 0000                | 0000 0000                                      |
| 15h                  | CCPR1L  | Capture/Compare/PWM Register1 (LSB)  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 16h                  | CCPR1H  | Capture/Compare/PWM Register1 (MSB)  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 17h                  | CCP1CON | —  | —       | CCP1X   | CCP1Y  | CCP1M3          | CCP1M2            | CCP1M1  | CCP1M0  | --00 0000                | --00 0000                                      |
| 18h                  | RCSTA   | SPEN   | RX9     | SREN  | CREN   | —               | FERR              | OERR    | RX9D    | 0000 -00x                | 0000 -00x                                      |
| 19h                  | TXREG   | USART Transmit Data Register   |         |   |  |                 |                   |         |         | 0000 0000                | 0000 0000                                      |
| 1Ah                  | RCREG   | USART Receive Data Register  |         |   |  |                 |                   |         |         | 0000 0000                | 0000 0000                                      |
| 1Bh                  | CCPR2L  | Capture/Compare/PWM Register2 (LSB)  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 1Ch                  | CCPR2H  | Capture/Compare/PWM Register2 (MSB)  |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 1Dh                  | CCP2CON | —  | —       | CCP2X   | CCP2Y  | CCP2M3          | CCP2M2            | CCP2M1  | CCP2M0  | --00 0000                | --00 0000                                      |
| 1Eh                  | ADRES   | A/D Result Register Byte   |         |   |  |                 |                   |         |         | x00xx x00xx              | uuuu uuuu                                      |
| 1Fh                  | ADCON0  | ADCS1  | ADCS0   | CHS2  | CHS1   | CHS0            | $\frac{GO}{DONE}$ | —       | ADON    | 0000 00-0                | 0000 00-0                                      |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

- Note** 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.
- 2: Other (non power-up) RESETS include external RESET through MCLR and Watchdog Timer Reset.
- 3: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4: These registers can be addressed from any bank.
- 5: PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.
- 6: This bit always reads as a '1'.

Tabla 4. Registros de funciones especiales (continuación)

| Address              | Name           | Bit 7  | Bit 6  | Bit 5                         | Bit 4  | Bit 3           | Bit 2                     | Bit 1  | Bit 0  | Value on:<br>POR,<br>BOR | Value on<br>all other<br>RESETS <sup>(2)</sup> |           |
|----------------------|----------------|--|--------|-------------------------------|--|-----------------|---------------------------|--------|--------|--------------------------|--|-----------|
| Bank 1               |                |  |        |                               |  |                 |                           |        |        |                          |  |           |
| 80h <sup>(4)</sup>   | INDF           | Addressing this location uses contents of FSR to address data memory (not a physical register) |        |                               |  |                 |                           |        |        | 0000 0000                | 0000 0000                                      |           |
| 81h                  | OPTION_<br>REG | RBP $\overline{U}$   | INTEDG | T0CS                          | T0SE   | PSA             | PS2                       | PS1    | PS0    | 1111 1111                | 1111 1111                                      |           |
| 82h <sup>(4)</sup>   | PCL            | Program Counter's (PC) Least Significant Byte  |        |                               |  |                 |                           |        |        | 0000 0000                | 0000 0000                                      |           |
| 83h <sup>(4)</sup>   | STATUS         | IRP  | RP1    | RP0                           | $\overline{TO}$  | $\overline{PD}$ | Z                         | DC     | C      | 0001 1xxx                | 000q quuu                                      |           |
| 84h <sup>(4)</sup>   | FSR            | Indirect data memory address pointer   |        |                               |  |                 |                           |        |        | xxxx xxxx                | uuuu uuuu                                      |           |
| 85h                  | TRISA          | —  | —      | PORTA Data Direction Register |  |                 |                           |        |        | --11 1111                | --11 1111                                      |           |
| 86h                  | TRISB          | PORTB Data Direction Register  |        |                               |  |                 |                           |        |        | 1111 1111                | 1111 1111                                      |           |
| 87h                  | TRISC          | PORTC Data Direction Register  |        |                               |  |                 |                           |        |        | 1111 1111                | 1111 1111                                      |           |
| 88h <sup>(5)</sup>   | TRISD          | PORTD Data Direction Register  |        |                               |  |                 |                           |        |        | 1111 1111                | 1111 1111                                      |           |
| 89h <sup>(5)</sup>   | TRISE          | IBF  | OBF    | IOV                           | PSPMODE  | —               | PORTE Data Direction Bits |        |        | 0000 -111                | 0000 -111                                      |           |
| 8Ah <sup>(1,4)</sup> | PCLATH         | —  | —      | —                             | Write Buffer for the upper 5 bits of the Program Counter |                 |                           |        |        |                          | ---0 0000                                      | ---0 0000 |
| 8Bh <sup>(4)</sup>   | INTCON         | GIE  | PEIE   | T0IE                          | INTE   | RBIE            | T0IF                      | INTF   | RBIF   | 0000 000x                | 0000 000u                                      |           |
| 8Ch                  | PIE1           | PSP1E <sup>(3)</sup>   | ADIE   | RCIE                          | TXIE   | SSPIE           | CCP1IE                    | TMR2IE | TMR1IE | 0000 0000                | 0000 0000                                      |           |
| 8Dh                  | PIE2           | —  | —      | —                             | —  | —               | —                         | —      | CCP2IE | ---- -000                | ---- -000                                      |           |
| 8Eh                  | PCON           | —  | —      | —                             | —  | —               | —                         | POR    | BOR    | ---- -ggg                | ---- -uuu                                      |           |
| 8Fh                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 90h                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 91h                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 92h                  | PR2            | Timer2 Period Register   |        |                               |  |                 |                           |        |        | 1111 1111                | 1111 1111                                      |           |
| 93h                  | SSPADD         | Synchronous Serial Port (I <sup>2</sup> C mode) Address Register                               |        |                               |  |                 |                           |        |        | 0000 0000                | 0000 0000                                      |           |
| 94h                  | SSPSTAT        | SMP  | CKE    | D/A                           | P  | S               | R/W                       | UA     | BF     | 0000 0000                | 0000 0000                                      |           |
| 95h                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 96h                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 97h                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 98h                  | TXSTA          | CSRC   | TX9    | TXEN                          | SYNC   | —               | BRGH                      | TRMT   | TX9D   | 0000 -010                | 0000 -010                                      |           |
| 99h                  | SPBRG          | Baud Rate Generator Register   |        |                               |  |                 |                           |        |        | 0000 0000                | 0000 0000                                      |           |
| 9Ah                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 9Bh                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 9Ch                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 9Dh                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 9Eh                  | —              | Unimplemented  |        |                               |  |                 |                           |        |        | —                        | —  |           |
| 9Fh                  | ADCON1         | —  | —      | —                             | —  | —               | PCFG2                     | PCFG1  | PCFG0  | ---- -000                | ---- -000                                      |           |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved.

Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** Other (non power-up) RESETS include external RESET through MCLR and Watchdog Timer Reset.

**3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**4:** These registers can be addressed from any bank.

**5:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

**6:** This bit always reads as a '1'.

Tabla 4. Registros de funciones especiales (continuación)

| Address               | Name       | Bit 7  | Bit 6  | Bit 5                   | Bit 4  | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on:<br>POR,<br>BOR | Value on<br>all other<br>RESETS <sup>(2)</sup> |
|-----------------------|------------|--|--------|-------------------------|--|-------|-------|-------|-------|--------------------------|--|
| Bank 2                |            |  |        |                         |  |       |       |       |       |                          |  |
| 100h <sup>(4)</sup>   | INDF       | Addressing this location uses contents of FSR to address data memory (not a physical register) |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |
| 101h                  | TMR0       | Timer0 Module's Register   |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 102h <sup>(4)</sup>   | PCL        | Program Counter's (PC) Least Significant Byte  |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |
| 103h <sup>(4)</sup>   | STATUS     | IRP  | RP1    | RP0                     | TO   | PD    | Z     | DC    | C     | 0001 1xxxx               | 000q quuu                                      |
| 104h <sup>(4)</sup>   | FSR        | Indirect Data Memory Address Pointer   |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 105h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 106h                  | PORTB      | PORTB Data Latch when written: PORTB pins when read  |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 107h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 108h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 109h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 10Ah <sup>(1,4)</sup> | PCLATH     | —  | —      | —                       | Write Buffer for the upper 5 bits of the Program Counter |       |       |       |       | ---0 0000                | ---0 0000                                      |
| 10Bh <sup>(4)</sup>   | INTCON     | GIE  | PEIE   | TOIE                    | INTE   | RBIE  | TOIF  | INTF  | RBIF  | 0000 000x                | 0000 000u                                      |
| 10Ch                  | PMDATA     | Data Register Low Byte   |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 10Dh                  | PMADR      | Address Register Low Byte  |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 10Eh                  | PMDATH     | —  | —      | Data Register High Byte |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 10Fh                  | PMADRH     | —  | —      | —                       | Address Register High Byte                               |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| Bank 3                |            |  |        |                         |  |       |       |       |       |                          |  |
| 180h <sup>(4)</sup>   | INDF       | Addressing this location uses contents of FSR to address data memory (not a physical register) |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |
| 181h                  | OPTION_REG | RBPV   | INTEDG | T0CS                    | T0SE   | PSA   | PS2   | PS1   | PS0   | 1111 1111                | 1111 1111                                      |
| 182h <sup>(4)</sup>   | PCL        | Program Counter's (PC) Least Significant Byte  |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |
| 183h <sup>(4)</sup>   | STATUS     | IRP  | RP1    | RP0                     | TO   | PD    | Z     | DC    | C     | 0001 1xxxx               | 000q quuu                                      |
| 184h <sup>(4)</sup>   | FSR        | Indirect Data Memory Address Pointer   |        |                         |  |       |       |       |       | xxxx xxxx                | uuuu uuuu                                      |
| 185h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 186h                  | TRISB      | PORTB Data Direction Register  |        |                         |  |       |       |       |       | 1111 1111                | 1111 1111                                      |
| 187h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 188h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 189h                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 18Ah <sup>(1,4)</sup> | PCLATH     | —  | —      | —                       | Write Buffer for the upper 5 bits of the Program Counter |       |       |       |       | ---0 0000                | ---0 0000                                      |
| 18Bh <sup>(4)</sup>   | INTCON     | GIE  | PEIE   | TOIE                    | INTE   | RBIE  | TOIF  | INTF  | RBIF  | 0000 000x                | 0000 000u                                      |
| 18Ch                  | PMCON1     | __ (6)   | —      | —                       | —  | —     | —     | —     | RD    | 1--- ---0                | 1--- ---0                                      |
| 18Dh                  | —          | Unimplemented  |        |                         |  |       |       |       |       | —                        | —  |
| 18Eh                  | —          | Reserved maintain clear  |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |
| 18Fh                  | —          | Reserved maintain clear  |        |                         |  |       |       |       |       | 0000 0000                | 0000 0000                                      |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0', r = reserved.

Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** Other (non power-up) RESETS include external RESET through MCLR and Watchdog Timer Reset.

**3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

**4:** These registers can be addressed from any bank.

**5:** PORTD, PORTE, TRISD, and TRISE are not physically implemented on the 28-pin devices, read as '0'.

**6:** This bit always reads as a '1'.

### **2.5.4 Interrupciones**

Los PIC de la familia 16FX poseen hasta doce fuentes de interrupciones. El registro de control de interrupciones (INTCON) registra las solicitudes individuales de interrupciones en el bit bandera. Este registro cuenta además con un bit de habilitación de interrupción individual y uno de habilitación de interrupción global. El bit de habilitación de interrupción global (GIE) (INTCON<7>) habilita (si está seteado) todas las interrupciones no enmascaradas o deshabilita (si no está seteado) todas las interrupciones. Cuando el bit GIE está habilitado y un bit bandera y el bit de enmascaramiento están seteados, la interrupción será atendida inmediatamente. Las interrupciones individuales pueden ser deshabilitadas a través de sus correspondientes bits de habilitación en varios registros. Los bits de interrupciones individuales pueden ser seteados sin tener en cuenta el estado del bit GIE, este bit es borrado con el RESET.

La instrucción RETFIE termina la rutina de interrupción tan pronto como sea seteado el bit GIE, lo que rehabilita la interrupción.

Las banderas de interrupciones periféricas están contenidas en los registros de funciones especiales PIR1 y PIR2. Los correspondientes bit de habilitación se encuentran contenidas en los registros de funciones especiales PIE1 y PIE2 y el bit periférico de interrupción se encuentra en el registro INTCON

Cuando una interrupción es respondida el bit GIE se limpia para deshabilitar cualquier otra interrupción. La dirección de retorno es empujada dentro de la pila y el contador de programa es cargado con 0004h. Estando en la rutina de interrupción las interrupciones pueden ser determinadas sondeando los bits banderas de interrupción. Los bits banderas de interrupción deben ser limpiados en el software antes de rehabilitar las interrupciones para evitar posibles interrupciones recursivas.

### **2.5.5 Puertos de entrada salida**

El microcontrolador PIC16LF73 tiene tres puertos de entrada salida, el puerto **A**, el puerto **B** y el puerto **C**. Algunos pines de los puertos de entrada/salida se encuentran multiplexados para atender alguna otra función periférica. En general,

cuando alguna función periférica está habilitada el pin no puede usarse como una entrada/salida de propósito general. El puerto A posee seis bits de entrada salida y los puertos B y C cuentan con ocho cada uno.

Cada puerto tiene 3 registros para su operación:

- TRIS, controla la dirección de los datos.
- PORT, lee los niveles en los pines del puerto.
- ADCON1 (A/D Control Register), se usa para multiplexar los pines de entrada de digital a analógico.

Estableciendo a un '1' lógico cierto pin TRISA, por ejemplo del puerto A, haría este pin una entrada digital. Mientras que reseteándolo, lo haría un pin de salida. Operaciones de lectura o escritura sobre el registro PORT sería equivalente a una operación de lectura o escritura respectivamente sobre el puerto.

#### **2.5.6 Set de instrucciones**

Toda instrucción de los microcontroladores de la familia PIC16F7X es una palabra de 14 bits que está dividida en un OPCODE, que especifica el tipo de instrucción (ver Figura 2.7) y uno o más operandos que especifican la operación de la instrucción.



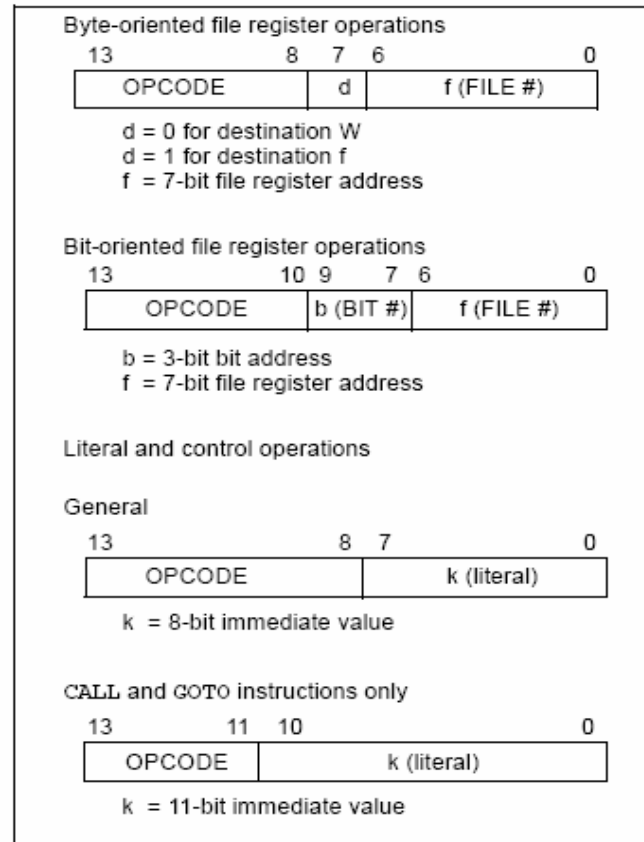


Figura 2.7 Formato de las instrucciones

El set de instrucciones esta dividido en tres categorías:

- Operaciones orientadas a bytes.
- Operaciones orientadas a bits.
- Operaciones literales.

En la tabla 5 se muestra la descripción del campo OPCODE.

Tabla 5 descripción del archivo OPCODE

| Field | Description  |
|-------|--|
| f     | Register file address (0x00 to 0x7F)   |
| w     | Working register (accumulator)   |
| b     | Bit address within an 8-bit file register  |
| k     | Literal field, constant data or label  |
| x     | Don't care location (= 0 or 1)<br>The assembler will generate code with x = 0.<br>It is the recommended form of use for compatibility with all Microchip software tools. |
| d     | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1   |
| PC    | Program Counter  |
| TO    | Time-out bit   |
| PD    | Power-down bit   |

Las instrucciones orientadas a bytes tienen 2 operandos:

1. El registro de archivo (especificado con 'f').
2. El destino del resultado (señalado con 'd').

La letra 'f' denota qué registro de archivo es usado en la instrucción. La designación 'd' especifica donde el resultado de la operación va a ser colocado. Si 'd' es cero el resultado es colocado en el registro W, si 'd' es uno el resultado es puesto en el registro archivo especificado en 'f'.

Todas las instrucciones orientadas a bit también contienen en sí dos operandos:

1. 'f' el registro de archivo.
2. 'b' el bit en el registro archivo.

El campo de bit designado por 'b' especifica el número del bit que será afectado por la operación. Mientras que 'f' especifica el registro archivo donde se encuentra este bit.

En las instrucciones literales 'k' representa una constante de ocho o nueve bits o también puede ser un valor literal.

La mayoría de las instrucciones tienen un solo ciclo de máquina (ver tabla 6) excepto cuando una prueba condicional resulta verdadera o cuando el contador de programa es cargado como resultado de una instrucción, en esos casos la instrucción tiene dos ciclos de máquina. Un ciclo de máquina está formado por cuatro períodos de oscilación con el cristal del micro de 4 MHz. El tiempo normal de duración de una instrucción es de 1  $\mu$ s. En el caso de las instrucciones de dos ciclos de máquina dura 2  $\mu$ s.

Tabla 6 Set de instrucciones del PIC16LF73

| Mnemonic,<br>Operands                  |      | Description                  | Cycles | 14-Bit Opcode |      |      |      | Status<br>Affected             | Notes |
|--|------|------------------------------|--------|---------------|------|------|------|--------------------------------|-------|
|  |      |                              |        | MSb           |      | LSb  |      |                                |       |
| BYTE-ORIENTED FILE REGISTER OPERATIONS |      |                              |        |               |      |      |      |                                |       |
| ADDWF                                  | f, d | Add W and f                  | 1      | 00            | 0111 | dfff | ffff | C,DC,Z                         | 1,2   |
| ANDWF                                  | f, d | AND W with f                 | 1      | 00            | 0101 | dfff | ffff | Z                              | 1,2   |
| CLRF                                   | f    | Clear f                      | 1      | 00            | 0001 | 1fff | ffff | Z                              | 2     |
| CLRWF                                  | -    | Clear W                      | 1      | 00            | 0001 | 0xxx | xxxx | Z                              |       |
| COMF                                   | f, d | Complement f                 | 1      | 00            | 1001 | dfff | ffff | Z                              | 1,2   |
| DECf                                   | f, d | Decrement f                  | 1      | 00            | 0011 | dfff | ffff | Z                              | 1,2   |
| DECFSZ                                 | f, d | Decrement f, Skip if 0       | 1(2)   | 00            | 1011 | dfff | ffff |                                | 1,2,3 |
| INCF                                   | f, d | Increment f                  | 1      | 00            | 1010 | dfff | ffff | Z                              | 1,2   |
| INCFSZ                                 | f, d | Increment f, Skip if 0       | 1(2)   | 00            | 1111 | dfff | ffff |                                | 1,2,3 |
| IORWF                                  | f, d | Inclusive OR W with f        | 1      | 00            | 0100 | dfff | ffff | Z                              | 1,2   |
| MOVF                                   | f, d | Move f                       | 1      | 00            | 1000 | dfff | ffff | Z                              | 1,2   |
| MOVWF                                  | f    | Move W to f                  | 1      | 00            | 0000 | 1fff | ffff |                                |       |
| NOP                                    | -    | No Operation                 | 1      | 00            | 0000 | 0xx0 | 0000 |                                |       |
| RLF                                    | f, d | Rotate Left f through Carry  | 1      | 00            | 1101 | dfff | ffff | C                              | 1,2   |
| RRF                                    | f, d | Rotate Right f through Carry | 1      | 00            | 1100 | dfff | ffff | C                              | 1,2   |
| SUBWF                                  | f, d | Subtract W from f            | 1      | 00            | 0010 | dfff | ffff | C,DC,Z                         | 1,2   |
| SWAPF                                  | f, d | Swap nibbles in f            | 1      | 00            | 1110 | dfff | ffff |                                | 1,2   |
| XORWF                                  | f, d | Exclusive OR W with f        | 1      | 00            | 0110 | dfff | ffff | Z                              | 1,2   |
| BIT-ORIENTED FILE REGISTER OPERATIONS  |      |                              |        |               |      |      |      |                                |       |
| BCF                                    | f, b | Bit Clear f                  | 1      | 01            | 00bb | bfff | ffff |                                | 1,2   |
| BSF                                    | f, b | Bit Set f                    | 1      | 01            | 01bb | bfff | ffff |                                | 1,2   |
| BTFSC                                  | f, b | Bit Test f, Skip if Clear    | 1 (2)  | 01            | 10bb | bfff | ffff |                                | 3     |
| BTFSS                                  | f, b | Bit Test f, Skip if Set      | 1 (2)  | 01            | 11bb | bfff | ffff |                                | 3     |
| LITERAL AND CONTROL OPERATIONS         |      |                              |        |               |      |      |      |                                |       |
| ADDLW                                  | k    | Add literal and W            | 1      | 11            | 111x | kkkk | kkkk | C,DC,Z                         |       |
| ANDLW                                  | k    | AND literal with W           | 1      | 11            | 1001 | kkkk | kkkk | Z                              |       |
| CALL                                   | k    | Call subroutine              | 2      | 10            | 0kkk | kkkk | kkkk |                                |       |
| CLRWDt                                 | -    | Clear Watchdog Timer         | 1      | 00            | 0000 | 0110 | 0100 | $\overline{TO}, \overline{PD}$ |       |
| GOTO                                   | k    | Go to address                | 2      | 10            | 1kkk | kkkk | kkkk |                                |       |
| IORLW                                  | k    | Inclusive OR literal with W  | 1      | 11            | 1000 | kkkk | kkkk | Z                              |       |
| MOVLW                                  | k    | Move literal to W            | 1      | 11            | 00xx | kkkk | kkkk |                                |       |
| RETFIE                                 | -    | Return from interrupt        | 2      | 00            | 0000 | 0000 | 1001 |                                |       |
| RETLW                                  | k    | Return with literal in W     | 2      | 11            | 01xx | kkkk | kkkk |                                |       |
| RETURN                                 | -    | Return from Subroutine       | 2      | 00            | 0000 | 0000 | 1000 |                                |       |
| SLEEP                                  | -    | Go into standby mode         | 1      | 00            | 0000 | 0110 | 0011 | $\overline{TO}, \overline{PD}$ |       |
| SUBLW                                  | k    | Subtract W from literal      | 1      | 11            | 110x | kkkk | kkkk | C,DC,Z                         |       |
| XORLW                                  | k    | Exclusive OR literal with W  | 1      | 11            | 1010 | kkkk | kkkk | Z                              |       |

**Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

**2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

**3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

## **2.6. Características técnicas del MRF24J40 Transmisor/Receptor.**

El MRF24J40 [3] es un transceptor compatible con la especificación IEEE 802.15.4-2003. Soporta el protocolo ZigBee, MiWi y otros protocolos propietarios. Este transmisor/receptor integra las capas PHY bandabase y la capa MAC en su arquitectura; puede funcionar en conjunto con un microprocesador sencillo para aplicar baja velocidad de datos a una multitud de aplicaciones inalámbricas. En la figura 2.8 se muestra el diagrama de distribución de pines del dispositivo.

Este dispositivo integra un receptor, un transmisor, un VCO y un PLL en un mismo circuito integrado. Usa además una arquitectura de radio avanzada para minimizar la cuenta de las partes externas y disminuir el consumo de potencia.

Consiste principalmente de TX/RX FIFOs, un controlador CSMA-CA, un constructor de supertramas, un filtro de tramas recibidas, una ingeniería de seguridad y un módulo de procesamiento de la señal digital.

Puede dividirse en cuatro grandes bloques funcionales:

- Un bus SPI que sirve como la interfase entre el microcontrolador y el MRF24J40.
- Registros de control que sirven para monitorizar y controlar el transceptor.
- Un módulo MAC que implementa la lógica de esta capa según el estándar IEEE 802.15.4-2003.
- Un manejador de la capa PHY que codifica y decodifica los datos analógicos.

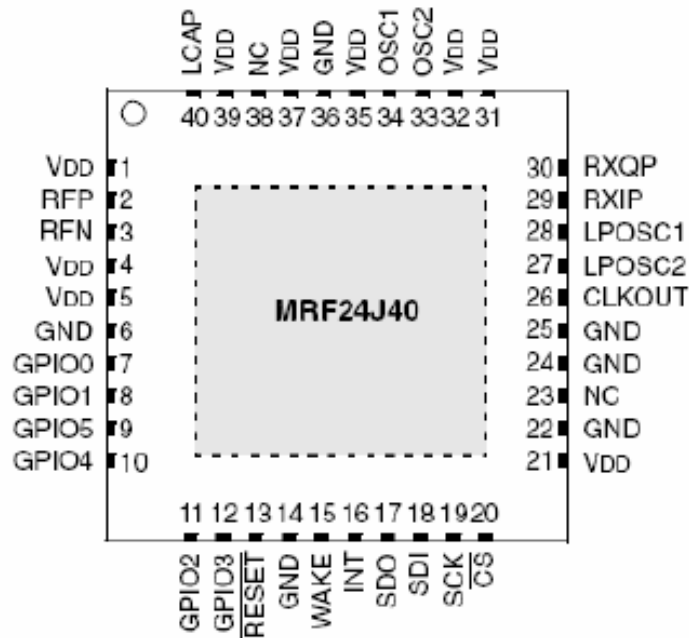


Figura 2.8 Distribución de pines del transceptor MRF24J40.

El dispositivo también contiene otros bloques, como el regulador de voltaje en el circuito, el módulo de seguridad y la lógica de control del sistema.

A continuación se presenta un resumen de las características del MRF24J40:

- Es compatible con la especificación IEEE 802.15.4-2003.
- Soporta MiWi, ZigBee y otros protocolos inalámbricos.
- Simple interface SPI de 4 líneas.
- Posee controladores integrados para los osciladores de 20 MHz y de 32.768 KHz.
- Posee una salida de reloj con frecuencia de 20 MHz disponible para ser proporcionada a un microcontrolador.
- Soporta modos de funcionamiento de bajo consumo de potencia.
- Bajos consumos de potencia, típicamente de 18 mA en recepción y 22 mA en transmisión.
- En modo "Sleep" el consumo es solo de 2 uA.
- La operación es en la banda ISM de 2.405-2.48 GHz.
- La sensibilidad típica es de -91dBm y el nivel de entrada máximo es de +5dBm.

- La potencia de salida normalmente es de 0 dBm, el rango de control de la potencia de TX es de 38.75 dBm.
- Entrada/salida de RF diferencial y un interruptor de TX/RX integrado.
- Mecanismos de hardware de CSMA-CA, respuesta ACK automática.
- Ingeniería de seguridad en hardware (AES-128).
- Capacidad de retransmisión de paquete automática.

En la tabla 7 que se presenta a continuación se muestra un resumen de sus principales características.

Tabla 7. Principales características del MRF24J40.

| Features                               | MRF24J40                                |
|--|---|
| IEEE 802.15.4™ Specification Compliant | Yes                                     |
| Integrated Oscillator Drive            | 20 MHz and 32.768 kHz                   |
| Reference Clock Output                 | 20 MHz                                  |
| Power-Saving Mode Support              | Yes                                     |
| Current Consumption                    | Typical 18 mA in RX and 22 mA in TX     |
| Sleep Mode                             | 2 $\mu$ A Typical                       |
| Serial Communications                  | SPI (4-wire)                            |
| Packages                               | 40-Pin Leadless QFN 6x6 mm <sup>2</sup> |

### 2.6.1 Interrupciones del MRF24J40

El MRF24J40 tiene una estructura de interrupción simple. Tiene un pin de interrupción que señala todos los eventos posibles. El registro ISRSTS es un registro de lectura y borrado que especifica cual es la interrupción que ha acaecido.

El registro INTMSK es usado para bloquear las fuentes de interrupciones no deseadas. El INTEDGE bit (CLKINTCR<1>) controla la polaridad de la interrupción. Una vez que el ISRSTS (registro de estado de interrupción) es leído por parte del microcontrolador, la bandera de interrupción se limpia. El microcontrolador está listo para manejar todas las banderas que retornan cada vez que el registro ISRSTS sea leído

## 2.6.2 Estructura de la interrupción

Cuando ocurre una interrupción habilitada, el terminal de interrupción permanecerá en su estado de interrupción, como ha sido determinado por INTEDGE bit hasta que todas las banderas que están causando la interrupción sean limpiadas o desenmascaradas (los bits de enmascaramiento están seteados) por el microcontrolador. Si más de una interrupción es habilitada, el microcontrolador puede encuestar cada bandera en el ISRSTS para determinar la fuente o las fuentes de interrupción.

Los registros asociados con las interrupciones del MRF24J40 se muestran en las figuras siguientes:

| RC-0  | RC-0   | RC-0      | RC-0  | RC-0 | RC-0     | RC-0     | RC-0  |
|-------|--------|-----------|-------|------|----------|----------|-------|
| SLPIF | WAKEIF | HSYMTMRIF | SECIF | RXIF | GTS2TXIF | GTS1TXIF | TXIF  |
| bit 7 |        |           |       |      |          |          | bit 0 |

|                   |                    |  |
|-------------------|--------------------|--|
| <b>Legend:</b>    | RC = Read to clear |  |
| R = Readable bit  | W = Writable bit   | U = Unimplemented bit, read as '0'           |
| -n = Value at POR | '1' = Bit is set   | '0' = Bit is cleared      x = Bit is unknown |

|       |  |
|-------|--|
| bit 7 | <b>SLPIF:</b> Sleep Alert Interrupt bit<br>1 = Sleep alert interrupt occurred<br>0 = Otherwise                   |
| bit 6 | <b>WAKEIF:</b> Wake-up Alert Interrupt bit<br>1 = Wake-up interrupt occurred<br>0 = Otherwise                    |
| bit 5 | <b>HSYMTMRIF:</b> Half Symbol Timer Interrupt bit<br>1 = Half symbol timer interrupt occurred<br>0 = Otherwise   |
| bit 4 | <b>SECIF:</b> Security Key Request Interrupt bit<br>1 = Security key request interrupt occurred<br>0 = Otherwise |
| bit 3 | <b>RXIF:</b> RX OK Interrupt bit<br>1 = RX OK interrupt occurred<br>0 = Otherwise                                |
| bit 2 | <b>GTS2TXIF:</b> GTS FIFO 2 Release Interrupt bit<br>1 = GTS2 transmission interrupt occurred<br>0 = Otherwise   |
| bit 1 | <b>GTS1TXIF:</b> GTS FIFO 1 Release Interrupt bit<br>1 = GTS1 transmission interrupt occurred<br>0 = Otherwise   |
| bit 0 | <b>TXIF:</b> TX FIFO Release Interrupt bit<br>1 = TX FIFO transmission interrupt occurred<br>0 = Otherwise       |

Figura 2.9 ISRSTS. Registro de estado de interrupciones

| R/W-1  | R/W-1   | R/W-1      | R/W-1  | R/W-1 | R/W-1     | R/W-1     | R/W-1 |
|--------|---------|------------|--------|-------|-----------|-----------|-------|
| SLPMSK | WAKEMSK | HSYMTMRMSK | SECMSK | RXMSK | GTS2TXMSK | GTS1TXMSK | TXMSK |
| bit 7  |         |            |        |       |           |           | bit 0 |

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

|       |  |
|-------|--|
| bit 7 | <b>SLPMSK:</b> Sleep Alert Mask bit<br>0 = Enable Sleep interrupt<br>1 = Otherwise                       |
| bit 6 | <b>WAKEMSK:</b> Wake-up Alert Mask bit<br>0 = Enable Wake interrupt<br>1 = Otherwise                     |
| bit 5 | <b>HSYMTMRMSK:</b> Half Symbol Timer Mask bit<br>0 = Enable half symbol timer interrupt<br>1 = Otherwise |
| bit 4 | <b>SECMSK:</b> Security Interrupt Mask bit<br>0 = Enable security interrupt<br>1 = Otherwise             |
| bit 3 | <b>RXMSK:</b> RX OK Mask bit<br>0 = Enable receive interrupt<br>1 = Otherwise                            |
| bit 2 | <b>GTS2TXMSK:</b> GTS FIFO 2 IRQ Mask bit<br>0 = Enable GTS FIFO 2 transmit interrupt<br>1 = Otherwise   |
| bit 1 | <b>GTS1TXMSK:</b> GTS FIFO 1 IRQ Mask bit<br>0 = Enable GTS FIFO 1 transmit interrupt<br>1 = Otherwise   |
| bit 0 | <b>TXMSK:</b> TX Normal FIFO IRQ Mask bit<br>0 = Enable normal FIFO transmit interrupt<br>1 = Otherwise  |

Figura 2.10 INTMSK. Registro de enmascaramiento de interrupciones

| U-0   | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0   | R/W-0    |
|-------|-----|-----|-----|-----|-----|---------|----------|
| —     | —   | —   | —   | —   | —   | INTEDGE | SLPCLKEN |
| bit 7 |     |     |     |     |     |         | bit 0    |

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

|         |  |
|---------|--|
| bit 7-2 | <b>Unimplemented:</b> Read as '0'  |
| bit 1   | <b>INTEDGE:</b> Interrupt Edge Polarity bit<br>1 = Rising edge<br>0 = Falling edge |
| bit 0   | <b>SLPCLKEN:</b> Sleep Clock Enable bit<br>1 = Disabled<br>0 = Enabled             |

Figura 2.11 CLKINTCR SLPCLK ON/OFF (habilitación del sleep del reloj) y registro de polaridad de interrupción.



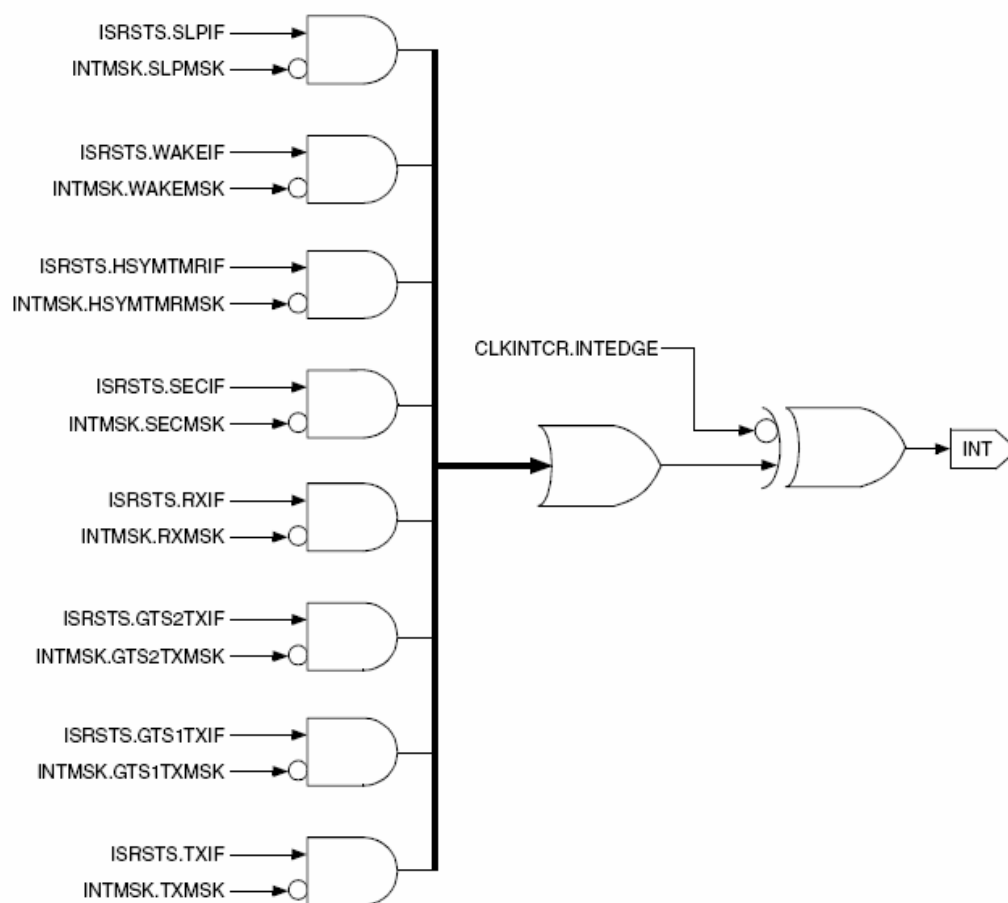


Figura 2.12 Lógica de interrupción del MRF24J40

### 2.6.3 Estructura del paquete IEEE 802.15.4

A continuación se expone a grueso modo la estructura de un paquete IEEE 802.15.4 [4]. El siguiente diagrama (figura 2.13) contiene detalles sobre la estructura de los paquetes que se transmiten/reciben con este chip.

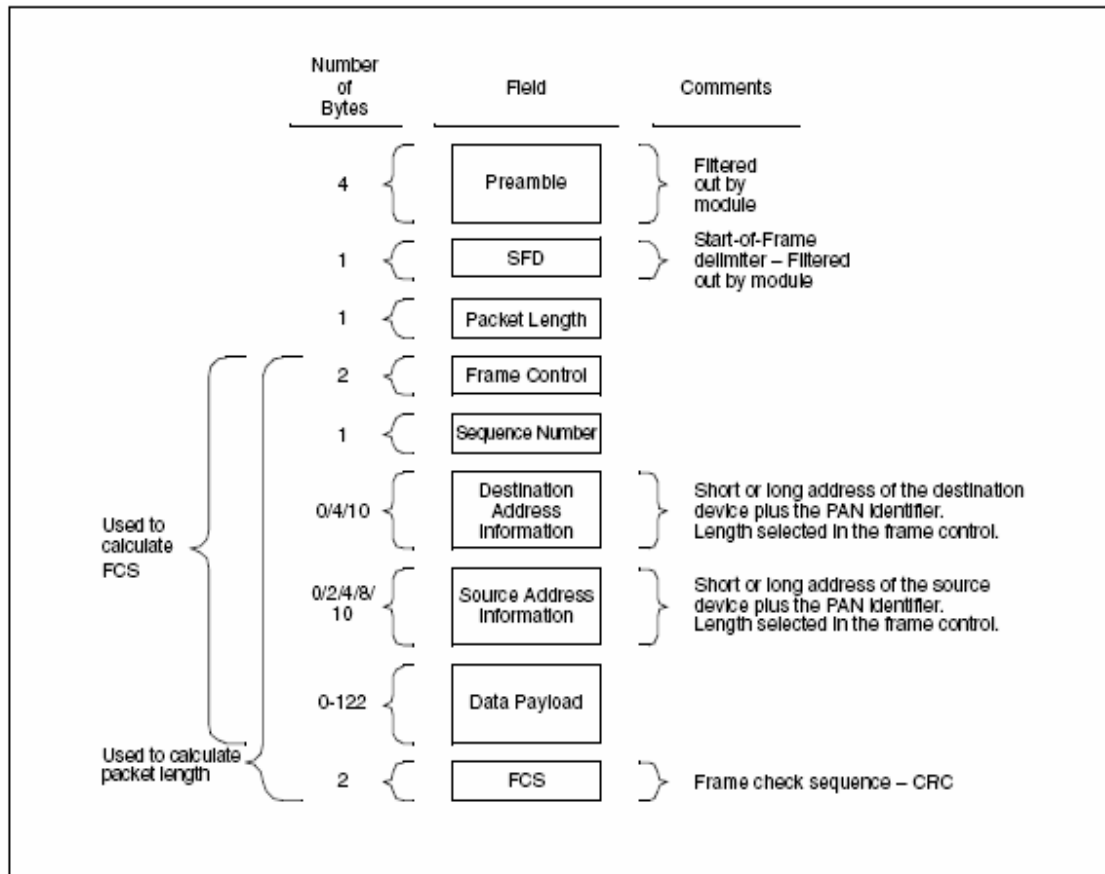


Figura 2.13. Campos del paquete IEEE 802.15.4

El bloque denotado como **preámbulo** en la figura es un delimitador del comienzo de la trama y es automáticamente generado de los paquetes cuando se transmite o se recibe.

Para el microcontrolador este campo es transparente. Normalmente, el PIC necesita conocer el campo CRC, el cual es generado automáticamente por el MRF24J40 en la transmisión y verificado en recepción. Este campo CRC es escrito directamente por el MRF24J40 en el buffer de recepción cuando el paquete ha arribado, el microcontrolador es el encargado de verificarlo.

**El campo longitud del paquete** especifica la cantidad de bytes que tiene el paquete, excluyendo el preámbulo y el SFD, pero incluyendo todos los otros campos incluso el FCS.

**El campo control de trama** especifica el tipo de paquete de que se trata, ya sea una trama de baliza, un ACK, o datos, etc. Este campo expresa los modos de direccionamiento usados, si el paquete está encriptado o no, si el paquete requiere ACK o si es enviado dentro de la misma red donde se generó o será enviado a otra red.

Esta información es usada por el PIC para determinar cómo descifrar los datos que siguen al campo control de trama.

**El campo número de secuencia** es de un byte y permite distinguir los paquetes. Este campo también es usado con los paquetes de reconocimiento ACK. Es bueno mencionar que el MRF24J40 tiene rasgos de autoreconocimiento en sí mismo.

**La información de la dirección de destino** puede cambiar en dependencia del campo control del tipo de trama del paquete. Este último puede detectar que ninguna dirección destino está presente, o que hay una dirección corta (2 bytes) o una dirección larga (8 bytes). En todos aquellos casos donde la dirección destino esté presente también deberá haber un identificador de la red PAN.

En los mensajes entrantes el MRF24J40 desechará automáticamente aquellos paquetes que no sean iguales a la configuración previa de la información de la radio. Esto elimina la necesidad de software para aquellos paquetes que no machean los requerimientos de direccionamiento.

Durante la transmisión se requiere que el microcontrolador escriba la dirección destino en el buffer de transmisión.

**La información de la dirección fuente** puede de igual forma cambiar en dependencia del campo de tipo de trama. Este último puede especificar que no existe una dirección fuente en el paquete, o que hay una dirección fuente corta de 2 bytes, o una dirección larga de 8 bytes.

El campo de control de trama puede decir también mediante el bit intra-PAN que la red destino es la misma donde se originó el paquete, en caso contrario este bit no se incluye.

Las direcciones largas consisten en dos porciones: los tres primeros bytes son conocidos como Identificador Único Organizacionalmente Extendido (EUI). Los identificadores son distribuidos por la IEEE 802.15.4. Los 5 bytes últimos son bytes de dirección que contienen los requerimientos necesarios a discreción de la compañía que administra su propio EUI.

Cuando se transmiten paquetes con sus direcciones fuentes largas o cortas dependiendo del bit campo de control de trama, deben ser escritas por el microcontrolador en el buffer de transmisión. El MRF24J40 no incluye automáticamente la información del campo de dirección fuente.

La sección del **paquete de datos** puede variar entre 0 y 122 bytes. Los paquetes que excedan los 127 bytes incluyendo el control de trama, direccionamiento fuente, direccionamiento destino, campos de datos y de FCS, serán filtrados o desechados por el MRF24J40.

**El campo FCS** es de 2 bytes y contiene datos estándar de la industria, y el CRC calculado con los datos del control de trama, número de secuencia, destino, fuente, y campo de datos.

Cuando se reciben paquetes el MRF24J40 verificará el campo CRC en cada uno. Si el bit RXCRCEN (activo a cero lógico) está a cero, y da inválida la suma de comprobación CRC el paquete se descarta. Pero si este bit está a '1' lógico y el paquete tiene correcto todos los otros requerimientos del filtrado, será escrito en el buffer de recepción y entonces es el microcontrolador quien debe comprobar la validez del campo CRC leyendo el vector de estado de recepción.

#### 2.6.4 Transmisión de paquetes

El control de acceso medio (MAC) en el MRF24J40 [3] generará automáticamente el preámbulo y el campo delimitador de comienzo de trama cuando se transmite. Adicionalmente el MAC puede generar cualquier PADDING (si es necesario) y el CRC si está configurado para hacerlo. El microcontrolador puede generar y escribir todos los otros campos de trama dentro de la memoria del buffer para la

transmisión. Antes de transmitir los paquetes, los registros del MAC que alteran las características de transmisión deben ser inicializados. En la figura 2.14 se muestra la estructura del paquete de transmisión.

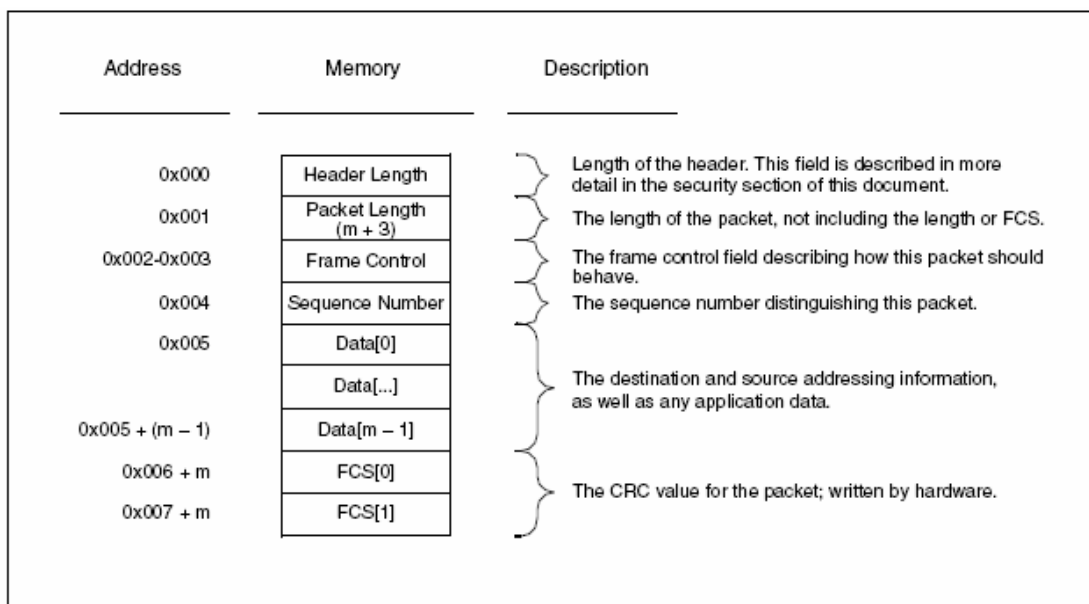


Figura 2.14 Diagrama del paquete de transmisión.

#### 2.6.4.1 Formato FIFO de transmisión

La capa MAC en la transmisión conforma tres tareas principales de tres maneras diferentes, de acuerdo a la norma IEEE 802.15.4:

- Control de transmisión FIFO.
- CSMA-CA automática y alineamiento del tiempo.
- Manejando la supertrama de hardware.

Para la función de control de transmisión FIFO la transmisión MAC controla cuatro FIFOs, incluyendo la baliza, normal y dos GTS FIFOs. Cuando todos los FIFOs son trigeriados, el TX MAC conforma un algoritmo CSMA-CA, envía un paquete a la transmisión bandabase (TXBB) en el momento correcto. Maneja la

retransmisión si se requiere un ACK y no es recibido y genera los bytes FCS automáticamente.

El algoritmo automático CSMA-CA conforma las alineaciones cronometradas tales como LIFS, SIFS y tiempo de respuesta ACK. El usuario puede simplemente programar los parámetros para el algoritmo CSMA-CA. La TX MAC actuará de acuerdo a esos parámetros.

Para el manejo de la supertrama de hardware, el TX MAC construirá el tiempo de trama de la supertrama, que incluye CAP, CFP INACTIVA y cada ranura de tiempo. El TX MAC envía los FIFOs: normal, baliza y GTS en el tiempo correcto automáticamente en cada transmisión. Esto reduce grandemente la complejidad del modo baliza habilitada de la norma IEEE 802.15.4.

#### **2.6.5 Recepción de paquetes**

Cuando el MRF24J40 recibe un paquete que pasa la capa MAC, el umbral y el filtro de tipo de paquete, le indicará la recepción al microcontrolador poniendo a uno "1" el bit RXIF, bit 3 del registro ISRSTS. El paquete permanecerá en el buffer hasta que el microcontrolador lo vacíe. Ningún otro paquete puede ser recibido mientras que el buffer no sea vaciado.

Cuando los paquetes pasan todos los filtros habilitados se coloca en el FIFO de recepción como se muestra en la figura 2.15.

En la figura 2.16 se muestra el diagrama de flujo del proceso de recepción.

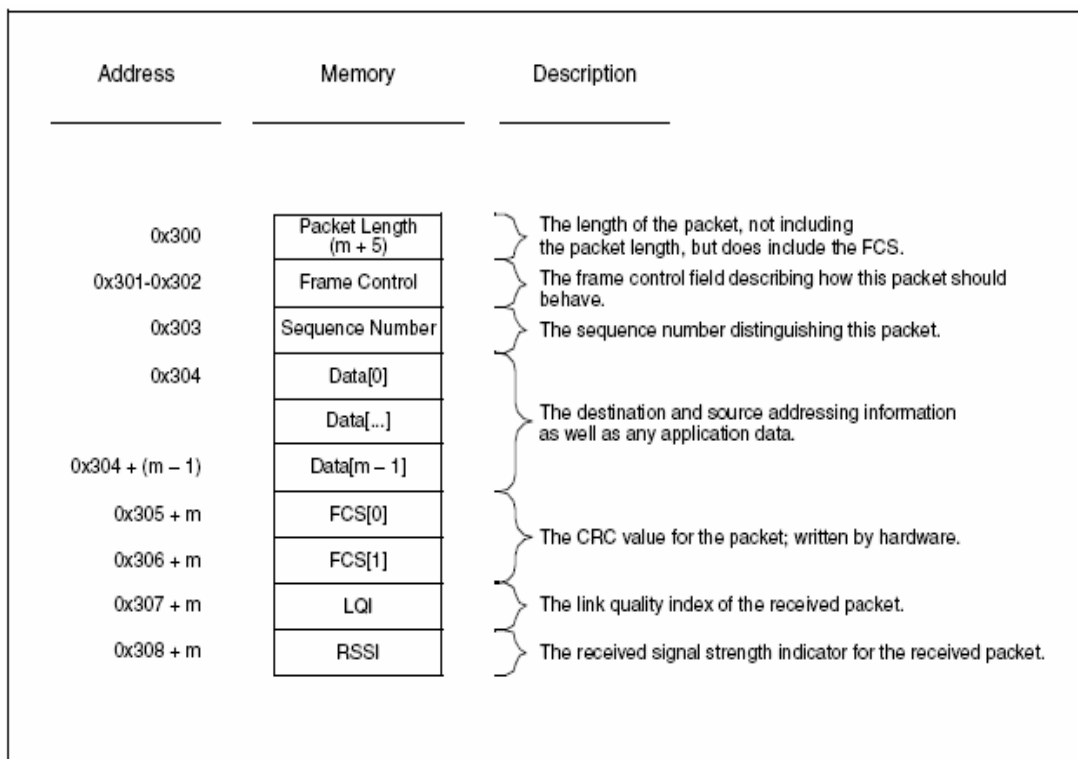


Figura 2.15 Diagrama del paquete de recepción.

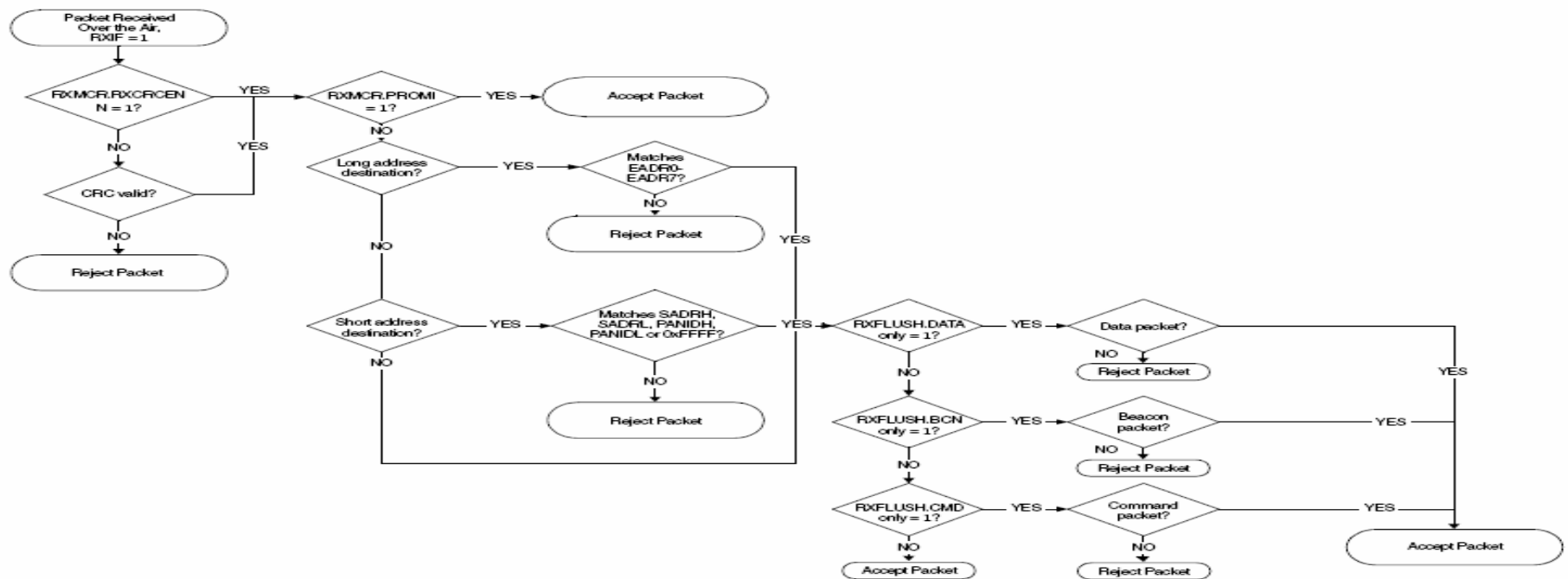


Figura 2.16. Diagrama de flujo del proceso de recepción.



### 2.6.6 Componentes periféricos del MRF24J40.

El MRF24J40 está diseñado para trabajar con un oscilador de cristal conectado a los pines OSC1 y OSC2, como se muestra en la figura 2.17.

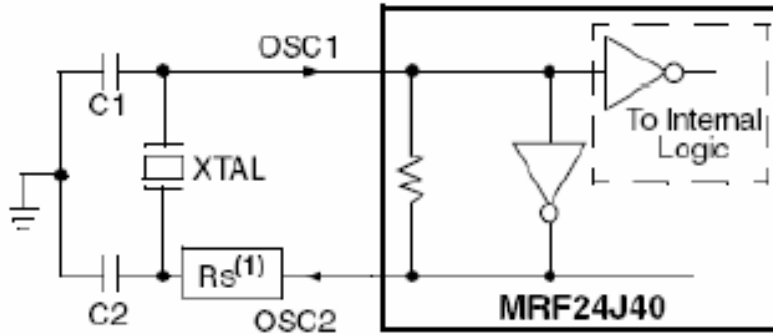


Figura 2.17 Oscilador de cristal del MRF24J40.

El pin CLKOUT se le provee al diseñador del sistema como una salida de reloj de 20 MHz para un microcontrolador anfitrión o para otros periféricos. También RFP y RFN son los pines de entrada/salida diferencial que van a la antena del sistema.

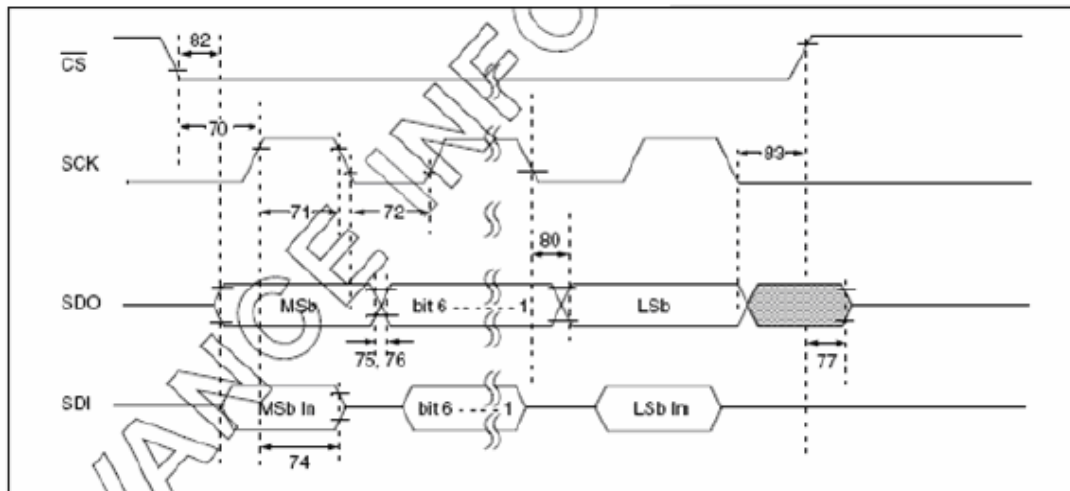


Figura 2.18 La comunicación SPI con el MRF24J40.

Los datos tienen su entrada al dispositivo a través del pin SDI (figura 2.18) Éstos son entradas bit a bit en cada flanco de subida del reloj por el pin SCK. Cuando el

MRF24J40 envía datos al PIC, esto se realiza por el pin SDO y en cada flanco de subida por el pin SCK. El pin CS debe permanecer a '0' mientras cualquiera de estas operaciones se realiza y luego devolverse a '1'.

### **2.6.7 Organización de la memoria del MRF 24J40**

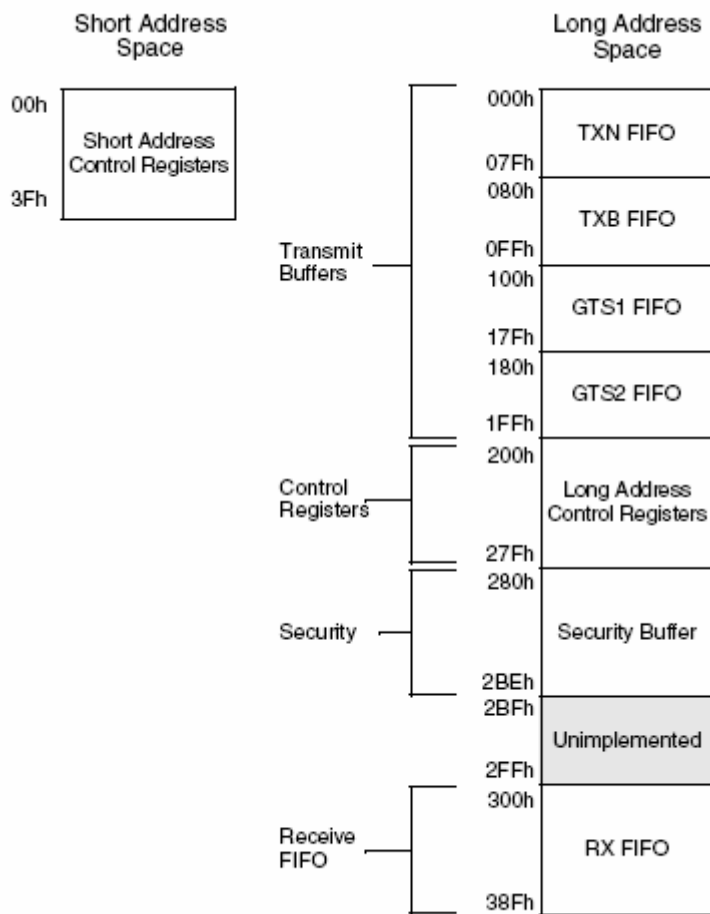
Toda la memoria en este dispositivo se implementa como una RAM estática. Contiene 5 tipos de memoria:

- Registros de control de direcciones cortas.
- Registros de control de direcciones largas.
- Buffers de transmisión.
- Buffers de recepción.
- Buffers de seguridad.

Los registros de control, ya sean de direcciones largas o cortas (figura 2.19) son usados para la configuración, el control y la retribución del estado del MRF24J40. Los registros de control son directamente leídos y escritos por medio de la interface SPI. Los buffers de transmisión y recepción contienen las memorias de transmisión y recepción que se usan para transmitir y recibir datos.

El buffer de seguridad provee una seguridad que es compatible con la especificación IEEE 802.15.4 LR-WPAN (ZigBee).

Todas estas memorias son leídas y escritas por medio de la interface SPI.



[6]

Figura 2.19 Registros de direcciones cortas y largas con sus direcciones.

A continuación se adjuntan los mapas de los registros de direcciones cortas y largas en las figuras 2.20 y 2.21. En la tabla 8 se muestra la descripción de los registros de direcciones cortas y en la tabla 9 la descripción de los registros de direcciones largas.

|     |         |     |          |     |      |     |           |
|-----|---------|-----|----------|-----|------|-----|-----------|
| 00h | RXMCR   | 10h | —        | 20h | —    | 30h | —         |
| 01h | PANIDL  | 11h | —        | 21h | —    | 31h | ISRSTS    |
| 02h | PANIDH  | 12h | —        | 22h | —    | 32h | INTMSK    |
| 03h | SADRLL  | 13h | —        | 23h | —    | 33h | GPIO      |
| 04h | SADRH   | 14h | —        | 24h | TXSR | 34h | TRISGPIO  |
| 05h | EADR0   | 15h | —        | 25h | —    | 35h | —         |
| 06h | EADR1   | 16h | —        | 26h | —    | 36h | RFCTL     |
| 07h | EADR2   | 17h | —        | 27h | —    | 37h | —         |
| 08h | EADR3   | 18h | —        | 28h | —    | 38h | —         |
| 09h | EADR4   | 19h | —        | 29h | —    | 39h | —         |
| 0Ah | EADR5   | 1Ah | —        | 2Ah | —    | 3Ah | BBREG2    |
| 0Bh | EADR6   | 1Bh | TXNMTIRG | 2Bh | —    | 3Bh | —         |
| 0Ch | EADR7   | 1Ch | —        | 2Ch | —    | 3Ch | —         |
| 0Dh | RXFLUSH | 1Dh | —        | 2Dh | —    | 3Dh | —         |
| 0Eh | —       | 1Eh | —        | 2Eh | —    | 3Eh | BBREG6    |
| 0Fh | —       | 1Fh | —        | 2Fh | —    | 3Fh | RSSITHCCA |

Figura 2.20 Mapa de los registros de direcciones cortas.

|      |         |      |          |      |         |      |   |      |   |
|------|---------|------|----------|------|---------|------|---|------|---|
| 200h | RFCTRL0 | 210h | —        | 220h | CLKCTRL | 230h | — | 240h | — |
| 201h | —       | 211h | CLKINTCR | 221h | —       | 231h | — | 241h | — |
| 202h | RFCTRL2 | 212h | —        | 222h | —       | 232h | — | 242h | — |
| 203h | RFCTRL3 | 213h | —        | 223h | —       | 233h | — | 243h | — |
| 204h | —       | 214h | —        | 224h | —       | 234h | — | 244h | — |
| 205h | —       | 215h | —        | 225h | —       | 235h | — | 245h | — |
| 206h | RFCTRL6 | 216h | —        | 226h | —       | 236h | — | 246h | — |
| 207h | RFCTRL7 | 217h | —        | 227h | —       | 237h | — | 247h | — |
| 208h | RFCTRL8 | 218h | —        | 228h | —       | 238h | — | 248h | — |
| 209h | —       | 219h | —        | 229h | —       | 239h | — | 249h | — |
| 20Ah | —       | 21Ah | —        | 22Ah | —       | 23Ah | — | 24Ah | — |
| 20Bh | —       | 21Bh | —        | 22Bh | —       | 23Bh | — | 24Bh | — |
| 20Ch | —       | 21Ch | —        | 22Ch | —       | 23Ch | — | 24Ch | — |
| 20Dh | —       | 21Dh | —        | 22Dh | —       | 23Dh | — |      |   |
| 20Eh | —       | 21Eh | —        | 22Eh | —       | 23Eh | — |      |   |
| 20Fh | —       | 21Fh | —        | 22Fh | —       | 23Fh | — |      |   |

Figura 2.21 Mapa de los registros direcciones largas.

Tabla 8. Descripción de los registros de direcciones cortas.

| File Name | Bit 7                                      | Bit 6   | Bit 5         | Bit 4    | Bit 3    | Bit 2     | Bit 1     | Bit 0   | Value on POR | Details on page: |
|-----------|--|---------|---------------|----------|----------|-----------|-----------|---------|--------------|------------------|
| RXMCR     | TXCRGEN                                    | BBLPBK  | ACKEN         | MACLPBK  | PANCOORD | COORD     | RXCRGEN   | PROMI   | 0000 0000    | 21               |
| PANIDL    | MAC PAN Low Byte (PANL<7:0>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| PANIDH    | MAC PAN High Byte (PANH<15:8>)             |         |               |          |          |           |           |         | 0000 0000    | 26               |
| SADRL     | MAC Short Address Low Byte (SADDRL<7:0>)   |         |               |          |          |           |           |         | 0000 0000    | 27               |
| SADRH     | MAC Short Address High Byte (SADDRH<15:8>) |         |               |          |          |           |           |         | 0000 0000    | 27               |
| EADR0     | LSB of EUI (EADR0<7:0>)                    |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR1     | Byte 2 of EUI (EADR1<15:8>)                |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR2     | Byte 3 of EUI (EADR2<23:16>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR3     | Byte 4 of EUI (EADR3<31:24>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR4     | Byte 5 of EUI (EADR4<39:32>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR5     | Byte 6 of EUI (EADR5<47:40>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR6     | Byte 7 of EUI (EADR6<55:48>)               |         |               |          |          |           |           |         | 0000 0000    | 26               |
| EADR7     | MSB of EUI (EADR7<63:56>)                  |         |               |          |          |           |           |         | 0000 0000    | 26               |
| RXFLUSH   | —  | r       | r             | RXWRTBLK | CMDONLY  | DATAONLY  | BCNONLY   | RXFLUSH | ~000 0000    | 34               |
| TXNMTRIG  | —  | —       | —             | PENDACK  | INDIRECT | ACKREQ    | SECEN     | TXRTS   | ---0 0000    | 30               |
| TXSR      | TXRETRY<7:6>                               |         | CCAFAIL       | r        | r        | r         | r         | r       | 0000 0000    | 31               |
| ISRSTS    | SLPIF                                      | WAKEIF  | HSYMTMRIF     | SECIF    | RXIF     | GTS2TXIF  | GTS1TXIF  | TXIF    | 0000 0000    | 36               |
| INTMSK    | SLPMSK                                     | WAKEMSK | HSYMTMRMSK    | SECMASK  | RXMSK    | GTS2TXMSK | GTS1TXMSK | TXMSK   | 1111 1111    | 37               |
| GPIO      | —  | —       | GPIO5         | GPIO4    | GPIO3    | GPIO2     | GPIO1     | GPIO0   | --00 0000    | 39               |
| TRISGPIO  | —  | —       | TRISGP5       | TRISGP4  | TRISGP3  | TRISGP2   | TRISGP1   | TRISGP0 | --00 0000    | 40               |
| RFCTL     | r  | —       | —             | r        | r        | RFRST     | r         | r       | 0--0 0000    | 24               |
| BBREG2    | CCAMODE<7:6>                               |         | CCATHRES<5:2> |          |          |           | —         | —       | 0000 00--    | 25               |
| BBREG6    | RSSIREQ                                    | RXRSSI  | r             | r        | r        | r         | r         | RSSIROY | 0000 0001    | 25               |
| RSSITHCCA | RSSITHRES<7:0>                             |         |               |          |          |           |           |         | 0000 0000    | 23               |

Legend: — = unimplemented, r = reserved. Shaded cells are unimplemented, read as '0'.

Tabla 9. Descripción de los registros de direcciones largas.

| File Name | Bit 7        | Bit 6 | Bit 5  | Bit 4        | Bit 3    | Bit 2 | Bit 1       | Bit 0     | Value on POR | Details on page: |
|-----------|--------------|-------|--------|--------------|----------|-------|-------------|-----------|--------------|------------------|
| RFCTRL0   | CHANNEL<7:4> |       |        |              | —        | —     | —           | —         | 0000 ----    | 24               |
| RFCTRL2   | RFPLL        | r     | r      | r            | r        | —     | —           | —         | 0000 0---    | 22               |
| RFCTRL3   | TXPOWER<7:3> |       |        |              |          | —     | —           | —         | 0000 0---    | 22               |
| RFCTRL6   | TXFIL        | —     | r      | r            | BATMONEN | —     | —           | —         | 0-00 0---    | 23               |
| RFCTRL7   | SLPCLK<7:6>  |       | —      | —            | —        | —     | CLKDIV<1:0> |           | 00-- --00    | 8                |
| RFCTRL8   | —            | —     | —      | RF_VCO       | —        | —     | —           | SLPCLKOUT | ---0 ---0    | 23               |
| CLKINTCR  | —            | —     | —      | —            | —        | —     | INTEDGE     | SLPCLKEN  | ---- --00    | 38               |
| CLKCTRL   | r            | —     | CLKOEN | SCLKDIV<4:0> |          |       |             |           | 0-00 0000    | 7                |

Legend: — = unimplemented, r = reserved. Shaded cells are unimplemented, read as '0'.

Por ejemplo, para acceder a los registros de dirección corta se debe enviar un cero como el primer bit de la transferencia SPI. Los siguientes 6 bits son la dirección del registro al cual se desea acceder. El último bit debe ser un cero indicando que se trata de una operación de lectura. Luego en el próximo ciclo de

reloj se enviará por SDO el bit más significativo del valor que contiene el registro seguido por los restantes bits.

El diagrama de la figura 2.22 se muestra la secuencia en el tiempo de la lectura de un registro de direcciones cortas.

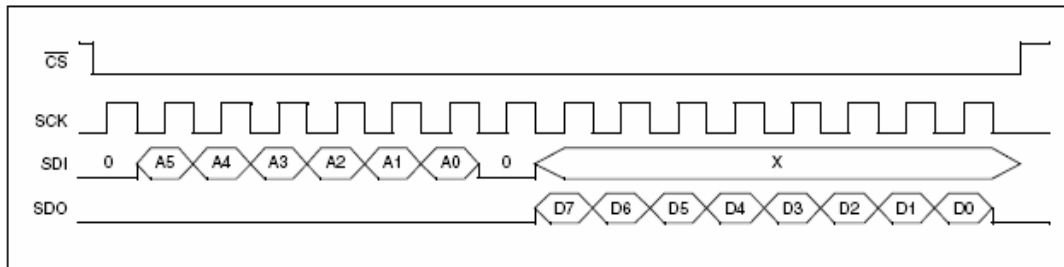


Figura 2.22 Diagrama en el tiempo de la lectura de un registro de dirección corta.

También se muestra un ejemplo en la figura 2.23 de cómo sería en código:

```

BYTE GetShortRAMAddress (BYTE address)
{
    BYTE toReturn;
    CSn = 0;
    SPIPut ((address<<1)&0b01111110);
    toReturn = SPIGet();
    CSn = 1;
    return toReturn;
}
    
```

Figura 2.23 Ejemplo de código para la lectura de un registro de dirección corta.

Ahora para la escritura de estos registros de dirección corta también se envía un '0' como el primer bit (este bit a cero indica precisamente que se trata de un acceso a un registro de dirección corta, cuando está a '1' indica un registro de dirección larga (ver figura 2.24).

Los siguientes 6 bits especifican la dirección del registro como tal. El bit final es un '1' para indicar que se trata de una operación de escritura del registro. Así, en el próximo ciclo de reloj se comienza a leer el nuevo valor que finalmente aparecerá en el registro.

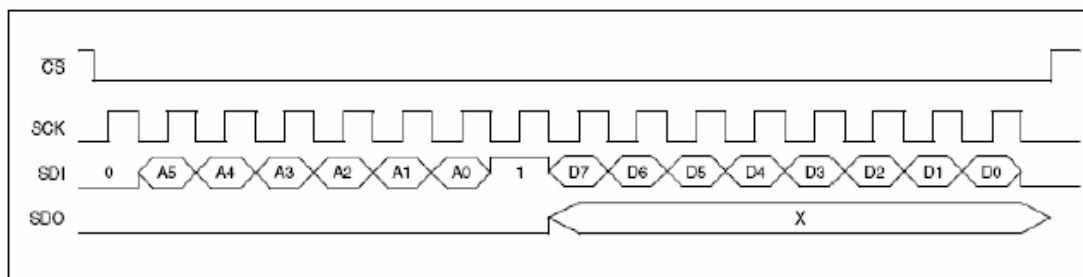


Figura 2.24 Diagrama en el tiempo de la escritura de un registro de dirección corta.

También se muestra un ejemplo de código para la escritura en registros de direcciones cortas, figura 2.25.

```
void SetShortRAMAddress(BYTE address, BYTE value)
{
    CSn = 0;
    SPIPut(((address<<1)&0b01111111)|0x01);
    SPIPut(value);
    CSn = 1;
}
```

Figura 2.25 Ejemplo de código para la escritura de un registro de dirección corta.

En el caso de una operación sobre un registro de dirección larga, el primer bit que indica de que tipo de registro se trata debe estar a '1' luego 10 bits que indican la dirección del registro y un bit final a '0' para lecturas y a '1' para escrituras. (Figura 2.26)

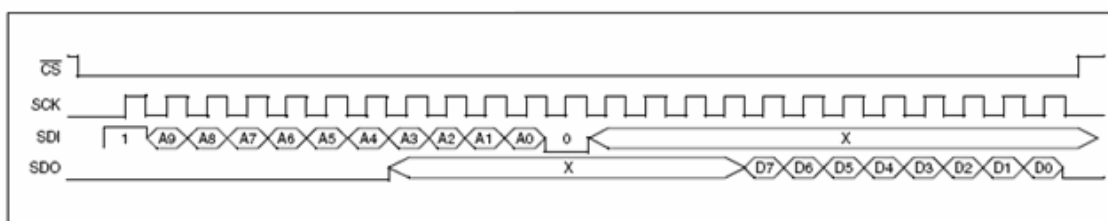


Figura 2.26 Diagrama en el tiempo de la lectura de un registro de dirección larga.

A continuación se muestra un ejemplo de código para la lectura de un registro de direcciones largas (figura 2.27).

```

BYTE GetLongRAMAddress(WORD address)
{
    BYTE toReturn;
    CSn = 0;
    SPIPut(((address>>3)&0b01111111)|0x80);
    SPIPut(((address<<5)&0b11100000));
    toReturn = SPIGet();
    CSn = 1;
    return toReturn;
}
    
```

Figura 2. 27 Ejemplo de código para la lectura de un registro de dirección larga.

En las dos figuras siguientes se muestran el diagrama en el tiempo de la escritura de un registro de direcciones largas (figura 2.28) y un ejemplo de código para la escritura de un registro de direcciones largas (figura 2.29)

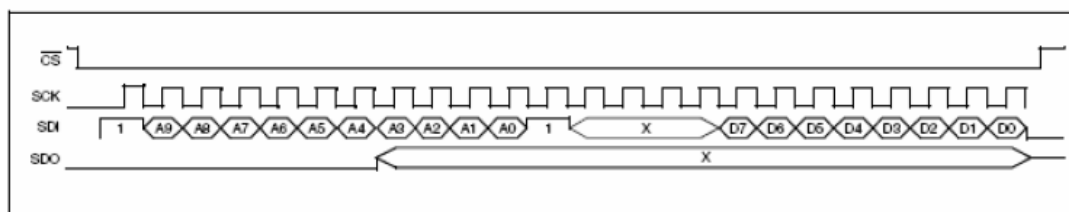


Figura 2.28 Diagrama en el tiempo de la escritura de un registro de dirección larga.

```

void SetLongRAMAddress(WORD address, BYTE value)
{
    CSn = 0;
    SPIPut(((BYTE)(address>>3)&0b01111111)|0x80);
    SPIPut(((BYTE)(address<<5)&0b11100000)|0x10);
    SPIPut(value);
    CSn = 1;
}
    
```

Figura 2.29. Ejemplo de código para la escritura de un registro de dirección larga.

A continuación se adjuntan algunas tablas con parámetros de operación de este transceptor:



Tabla 10. Condiciones de operación del MRF24J40.

| Parameters   | Min                   | Typ | Max                   | Units |
|--|-----------------------|-----|-----------------------|-------|
| Ambient Operating Temperature                      | -40                   | —   | +85                   | °C    |
| Supply Voltage for RF, Analog and Digital Circuits | 2.4                   | —   | 3.6                   | V     |
| Supply Voltage for Digital I/O                     | 2.4                   | 3.3 | 3.6                   | V     |
| Input High Voltage (V <sub>IH</sub> )              | 0.5 x V <sub>DD</sub> | —   | V <sub>DD</sub> + 0.3 | V     |
| Input Low Voltage (V <sub>IL</sub> )               | -0.3                  | —   | 0.2 x V <sub>DD</sub> | V     |

Tabla 11. Estados y consumos del MRF24J40.

Typical Values: T<sub>A</sub> = 25°C, V<sub>DD</sub> = 3.3V

| Chip Mode | Condition               | Min | Typ | Max | Units |
|-----------|-------------------------|-----|-----|-----|-------|
| Sleep     |                         | —   | 2   | TBD | μA    |
| TX        | At maximum output power | —   | 22  | TBD | mA    |
| RX        |                         | —   | 18  | TBD | mA    |

Legend: TBD = To Be Determined

### 2.6.8 Tarjeta de RF del MRF24J40 2.4 GHZ.

La tarjeta de radiofrecuencia de 2.4 GHZ (fig.2.30) que se utiliza en el proyecto usa el transceptor MRF24J40 de la Microchip que es compatible con la especificación IEEE 802.15.4 en la banda de los 2.4 GHZ. Esta tarjeta usa una antena trazada en el mismo circuito impreso, aunque puede ser fácilmente adaptada para una antena monopolo a conveniencia del usuario. En la figura 2.31 se muestra el esquema eléctrico de esta tarjeta.

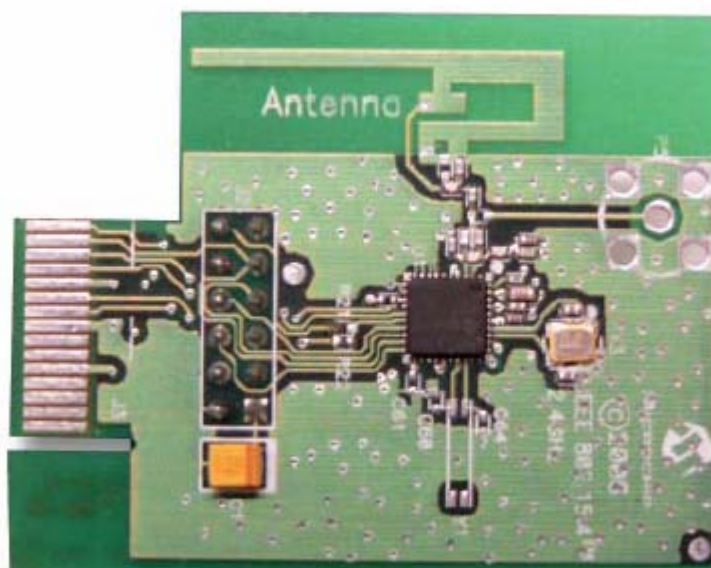


Figura 2.30 Tarjeta de Radiofrecuencia MRF24J40

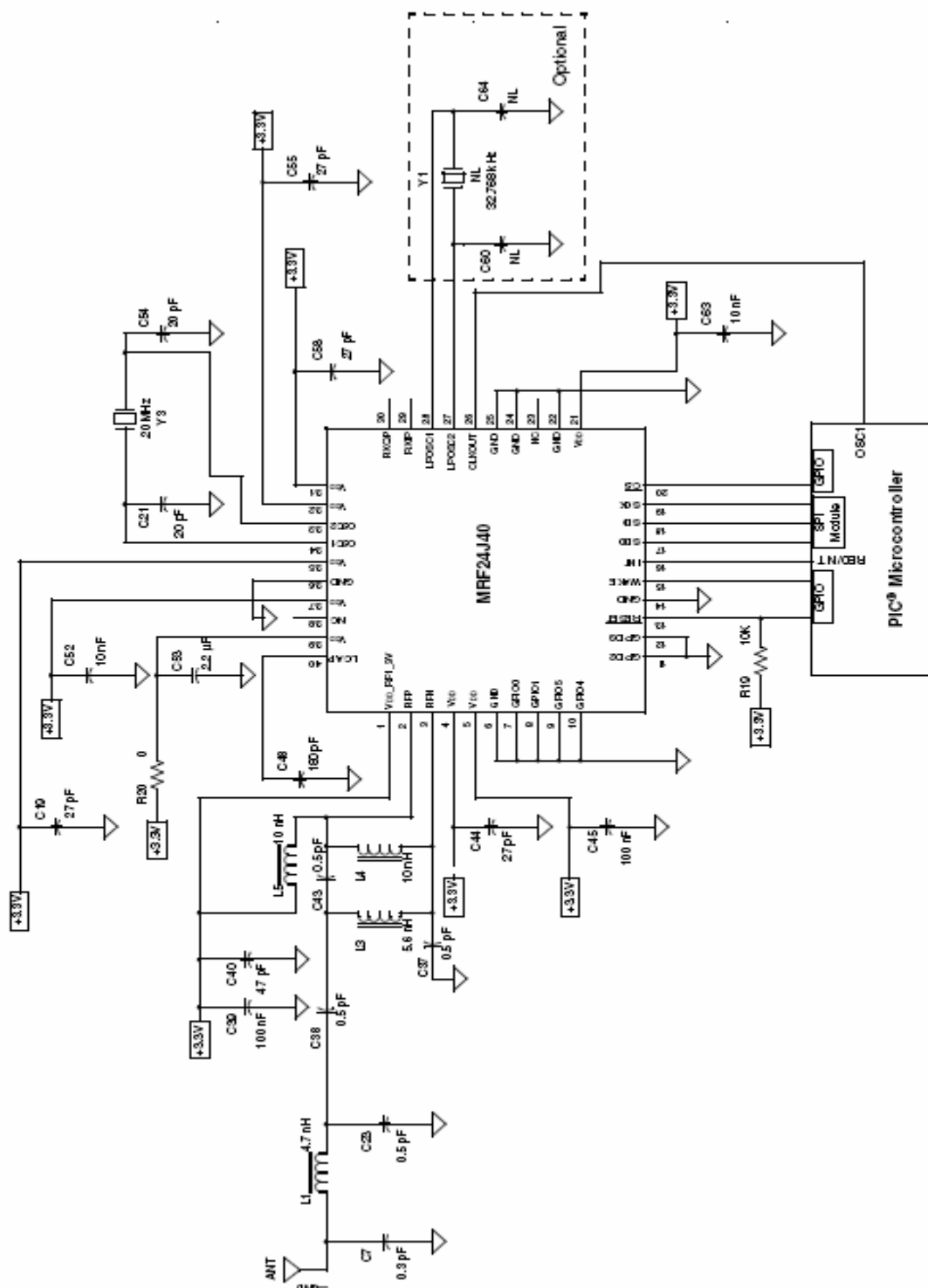


Figura 2.31 Circuito eléctrico de la tarjeta del MRF24J40

L5 es una bobina “choke” de RF, que filtra cualquier voltaje que no sea de 2.4 GHZ. L3, L4, C37 y C43 forman un “balun” que convierte la entrada diferencial no balanceada a una salida balanceada de un solo terminal y viceversa. L1, C23 y C38 forman un filtro tipo-pi que adapta la impedancia del “balun” a la impedancia de la antena. Otra parte importante en el hardware del MRF24J40 es la interfaz SPI para comunicarse con un microcontrolador. Esta interfaz está presente en la mayoría de los microcontroladores de la familia PIC16 y PIC18.

### 2.6.8.1 Características generales de la antena.

En la figura 2.32 se presenta el diagrama circuital del “balun” (tomado directamente del inglés). Éste es un adaptador de impedancia de la entrada no balanceada de la antena a la entrada balanceada de los pines RFP y RPN del transceptor.

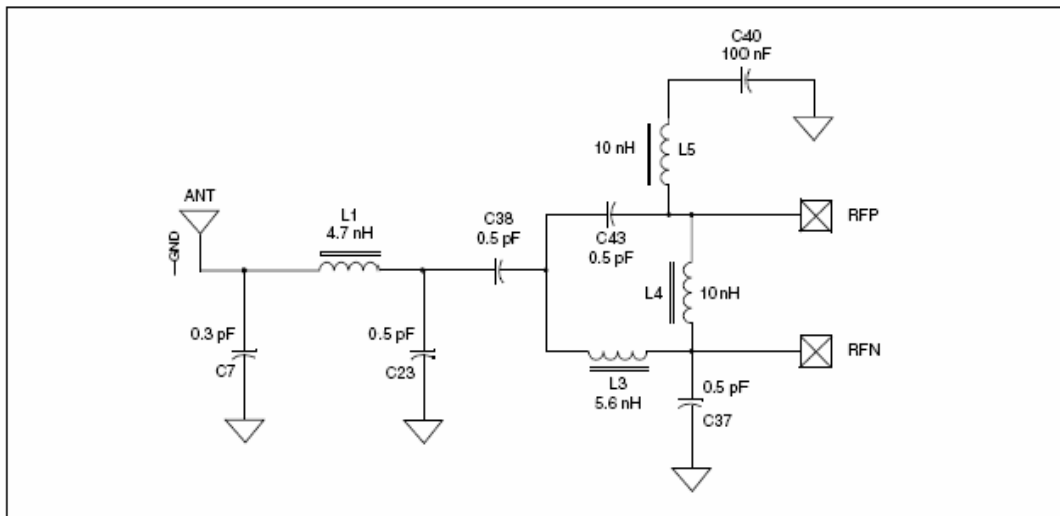


Figura 2.32. Esquema del acoplador de impedancia de la antena.

La figura 2.33 muestra un gráfico de las mediciones de la impedancia del balun cuando el centro de la banda está muy cercano a los 50Ω. Si se usan componentes de baja tolerancia (por ejemplo inferior al  $\pm 5\%$ ) con un buen aterramiento, la impedancia debe mantenerse cercana a este valor de 50Ω.

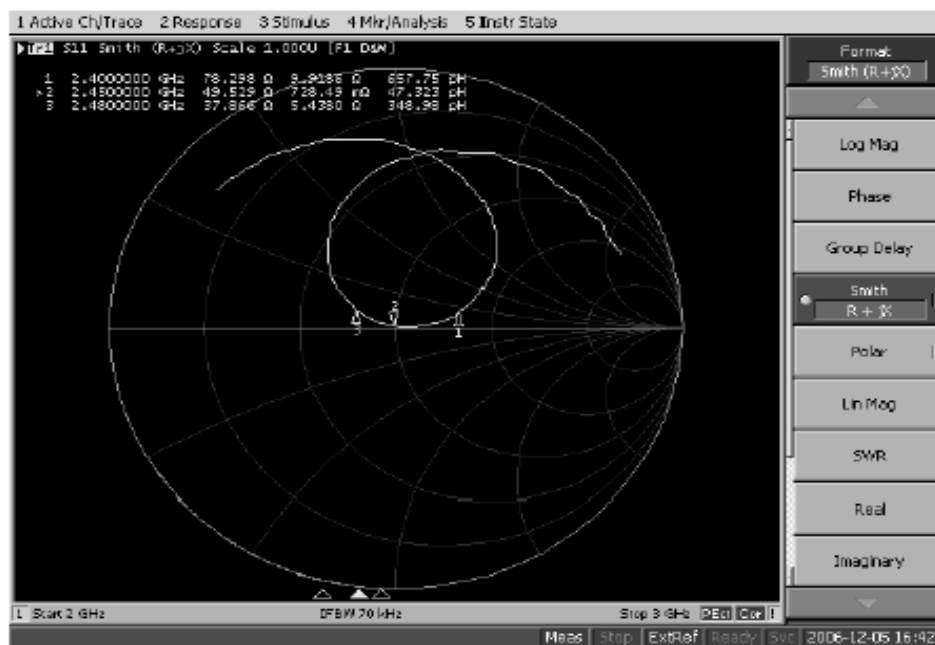


Figura 2.33 Gráfico del acoplador de impedancia.

Las figuras 2.34, 2.35 y 2.36 que siguen a continuación muestran la impedancia de la antena construida en el circuito impreso (topología PCB) en una escala logarítmica, en la carta de Smith y en niveles de voltaje, respectivamente.

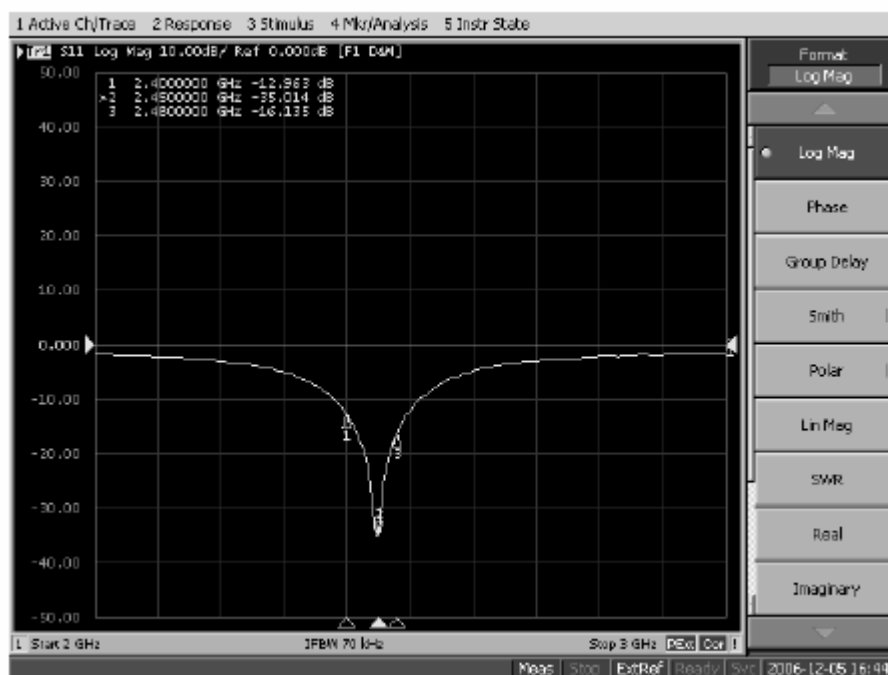


Figura 2.34. Impedancia de la antena PCB en escala logarítmica.

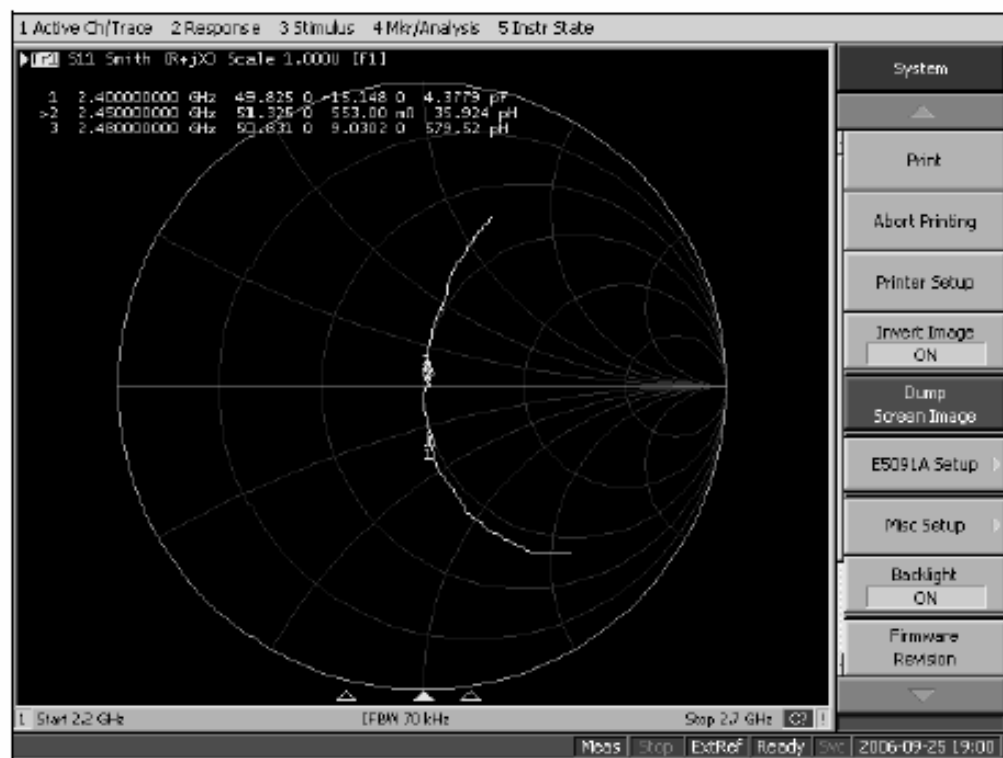


Figura 2.35. Representación en la carta de Smith de la antena.

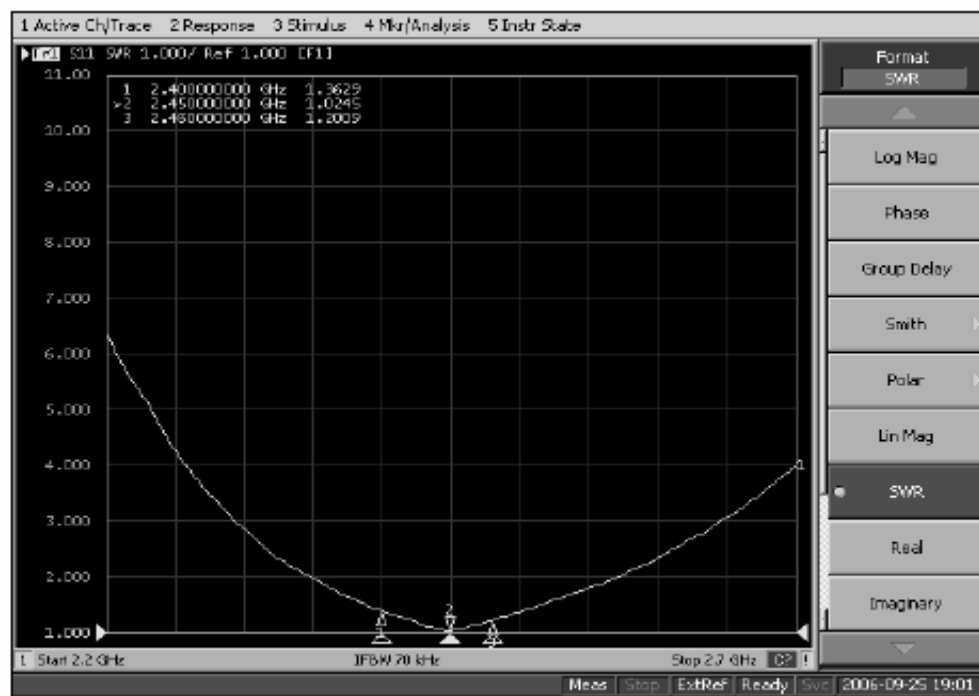


Figura 2.36. Niveles de voltaje en la antena.

Para el diseñador es una especificación muy crítica obtener fielmente los parámetros requeridos de impedancia para lo cual debe efectuar un eficiente diseño PCB, el cual aparece a modo de ejemplo en el fichero .pdf que contiene la hoja de datos del MRF24J40. Si las dimensiones presentadas a modo de ejemplo en el diseño que se presenta en dicha hoja de datos cambian por ejemplo en un 1mm, la impedancia de la antena en PCB cambiaría por lo menos 5-10 MHz.

## CAPÍTULO 3: Aspectos Prácticos del proyecto

### 3.1 Diseño y construcción del sistema de adquisición de datos

Aunque en el proyecto el SAD abarca una serie de parámetros, por consideraciones de tiempo y de recursos materiales disponibles se decidió trabajar solo con la variable tiempo (ver figura 3.1), ya que con un solo parámetro es posible cumplir los objetivos fundamentales del trabajo, que en este caso son realizar la comunicación entre el microcontrolador por el puerto SPI y el chip transceptor y realizar la comunicación entre el SAD y el lector, con lo que se lograría transmitir los datos almacenados en el primero hacia el segundo y de aquí a una PC, vía RS-232.



Figura 3.1. SAD

El esquema eléctrico del SAD diseñado con este fin se muestra en la figura 3.2. Como puede apreciarse en el esquema eléctrico, la alimentación se realiza mediante una fuente de 9V a partir de la cual se obtienen 3.3V con lo que se alimentan tanto el chip MRF24J40 como el PIC16LF73. se escogió este voltaje porque es el que recomienda el fabricante para el chip de comunicaciones.

Se utilizan dos relojes, uno de 32,768 kHz, que está conectado en los pines del temporizador cero, que sirve como base de tiempo para el conteo con precisión del tiempo de encendido del quemador de la kalfrisa, y otro de 4 MHz que se utiliza para el normal funcionamiento del programa, con lo que se logra la correcta comunicación entre el microcontrolador y el chip MRF24J40.

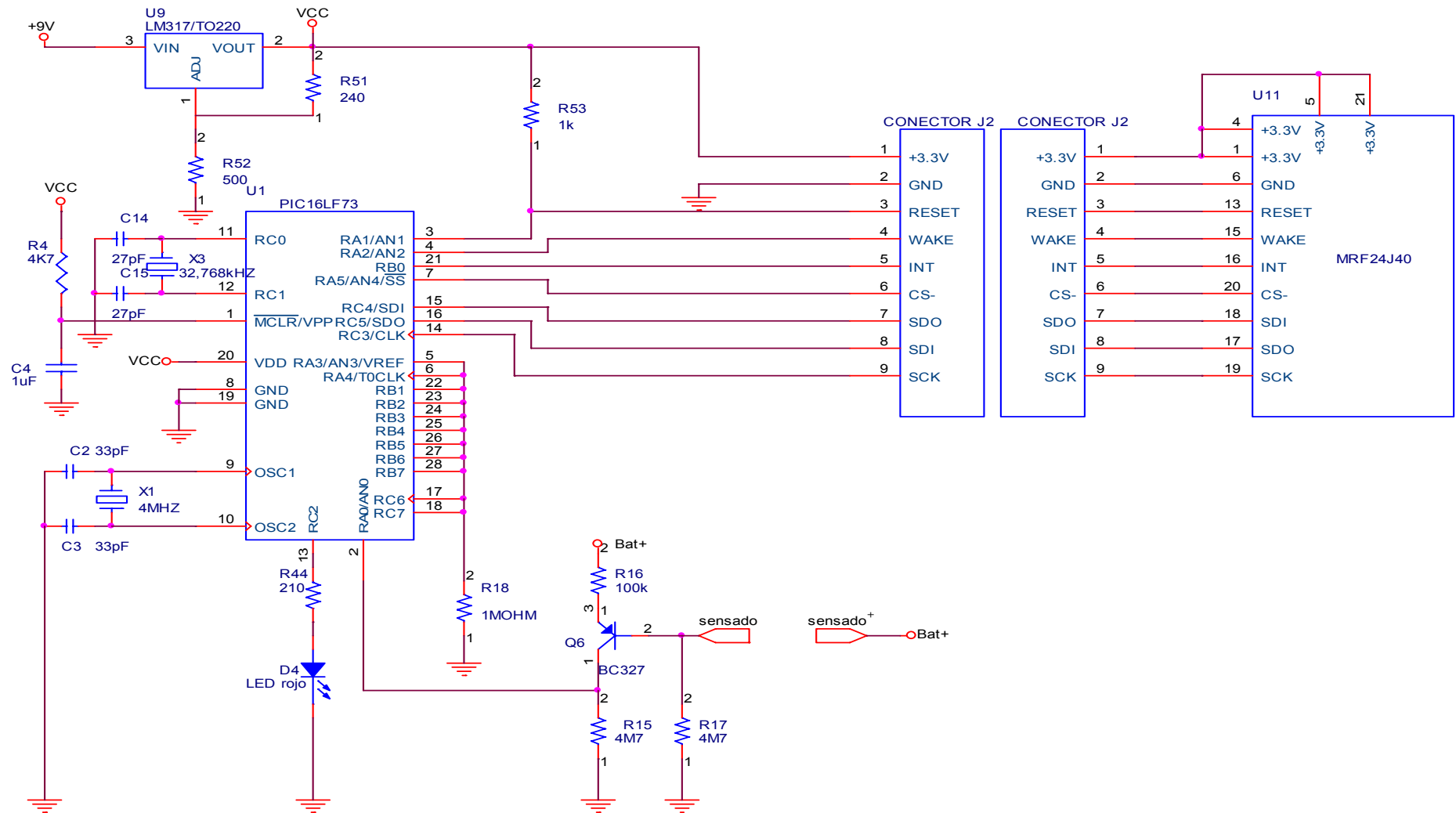


Figura 3.2. Esquema eléctrico del SAD



La comunicación SPI entre el microcontrolador y el MRF24J40 se realiza por los pines del micro reservados para esta función, ellos son:

RC3 Reloj de la SPI.

RC4 que cumple la función de entrada serie de datos (SDI).

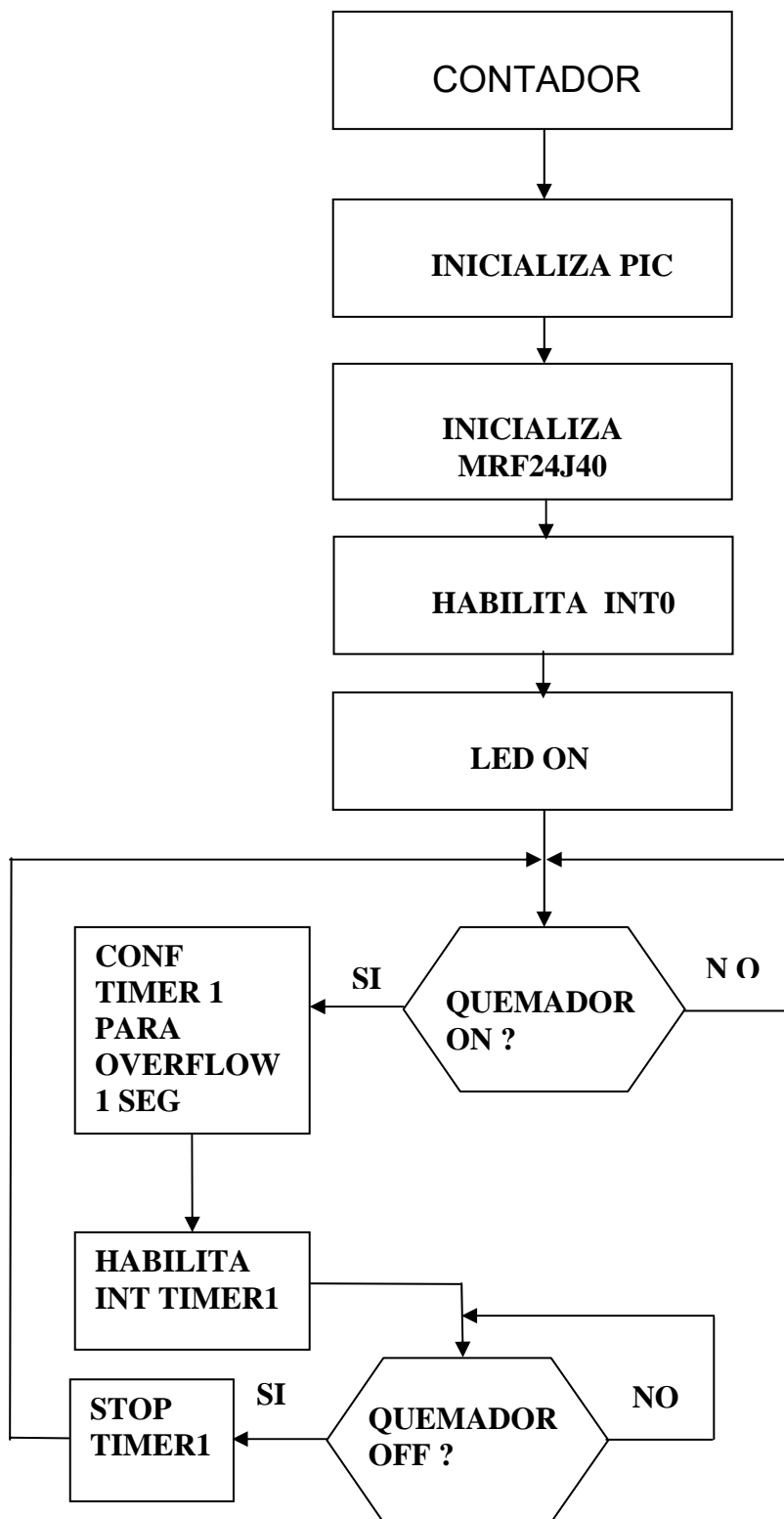
RC5 que cumple la función de salida serie de datos (SDO).

Además de las líneas del puerto SPI, se utilizan para el enlace con el chip otras salidas de puerto de propósito general que se utilizan para el RESET, WAKE, INT y CS.

El inicio del conteo del tiempo comienza cuando el Terminal RA0 pasa de un nivel bajo a un nivel alto, con esto se activa el temporizador cero que estará activado hasta tanto este terminal pase a un nivel bajo nuevamente.

### **3.2. Soluciones de Software para el SAD**

A continuación se muestra el diagrama de flujo del contador que muestra cómo se realiza la lectura del tiempo de trabajo del quemador.



La siguiente es una muestra de parte del código de inicialización del MRF24J40.

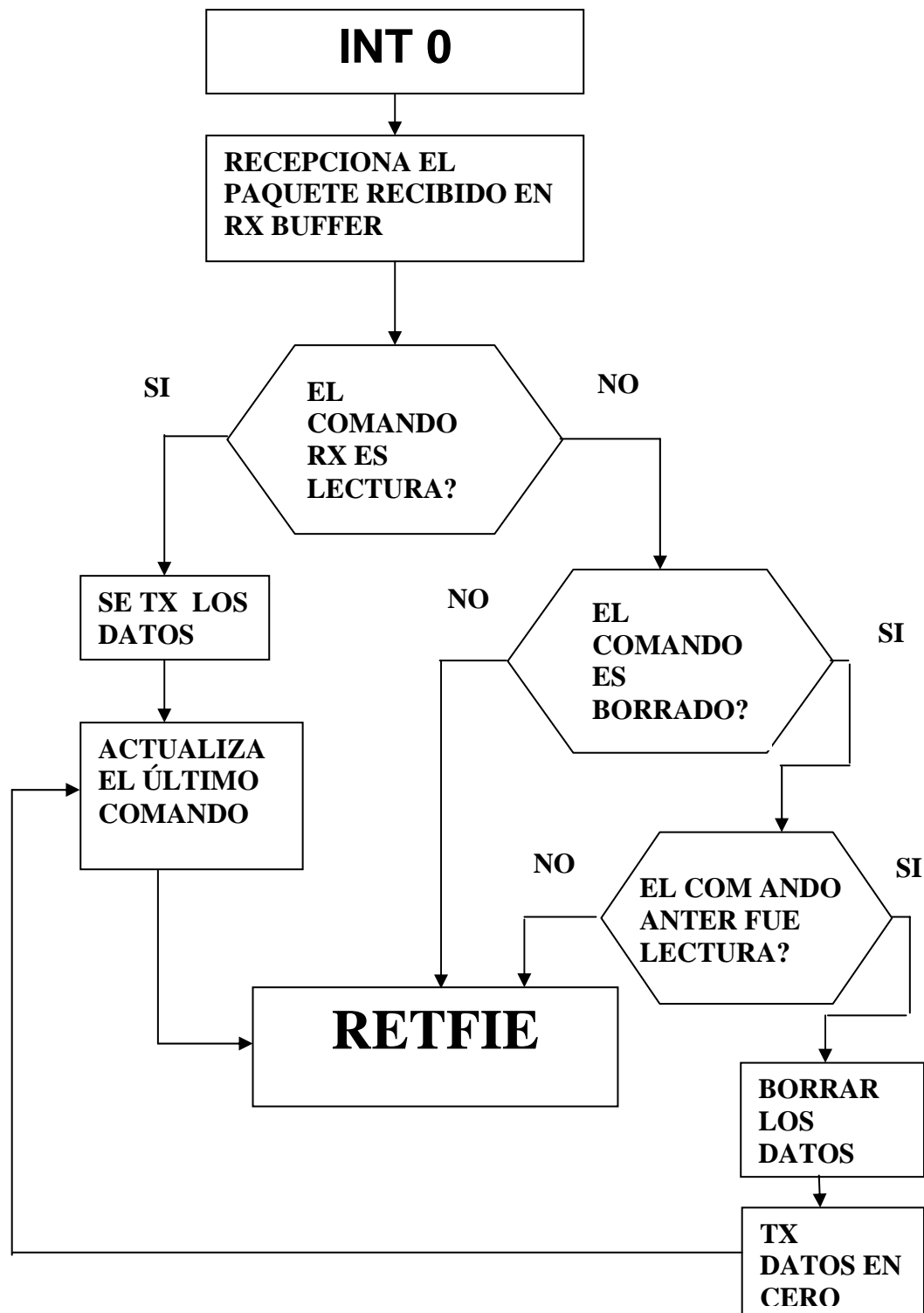
```
void MRF24J40Init(void)
{
    auto int i;
    auto long j;
    /* place the device in hardware reset */
    CHIPCON_RESET = 0;
    for(j=0;j<(long)300;j++){ }
    /* remove the device from hardware reset */
    CHIPCON_RESET = 1;
    for(j=0;j<(long)300;j++){ }
    /* reset the RF module */
    SetShortRAMAddress(RFCTL,0x04);
    /* remove the RF module from reset */
    SetShortRAMAddress(RFCTL,0x00);
    /* flush the RX fifo */
    SetShortRAMAddress(RXFLUSH,0x01);
    /* Program the short MAC Address, 0x0401 */
    SetShortRAMAddress(SADRL,0x01);
    SetShortRAMAddress(SADRH,0x04);
    SetShortRAMAddress(PANIDL,PANIDLOWB);
    SetShortRAMAddress(PANIDH,PANIDHIGHB);
    /* Program Long MAC Address*/
    for(i=0;i<(int)8;i++)
    {
        SetShortRAMAddress(EADR0+i,myLongAddress[i]);
    }
    /* program the RF and Baseband Register */
    SetLongRAMAddress(RFCTRL4,0x02);
    /* Habilita CRC automatica para la recepcion, ACKs */
    /* Descarta los paquetes con error de CRC y de direccion MAC */
    SetShortRAMAddress(RXMCR,0b00000010);
    /* Enable the RX */
    SetLongRAMAddress(RFRXCTRL,0x01);
```

```

/* setup */
SetLongRAMAddress(RFCTRL1,0x00);
/* enable the RF-PLL */
SetLongRAMAddress(RFCTRL2,0x80);
/* set TX for max output power */
SetLongRAMAddress(RFCTRL3,0x00);
/* enabled TX filter control */
SetLongRAMAddress(RFCTRL6,0x80);
SetLongRAMAddress(RFCTRL7,0b00000000);
SetLongRAMAddress(RFCTRL8,0b00010000);
/* Program CCA mode using RSSI */
SetShortRAMAddress(BBREG2,0x80);
/* Enable the packet RSSI */
SetShortRAMAddress(BBREG6,0x40);
/* Program CCA, RSSI threshold values */
SetShortRAMAddress(RSSITHCCA,0x60);
SetLongRAMAddress(RFCTRL0,0x00); //channel 11
/* FFD device */
SetShortRAMAddress(ACKTMOUT,0xB9);
/* Habilitar interrupcion por recepcion de un paquete valido */
SetShortRAMAddress(INTMSK,0b11110111);
// interrupcion por flanco de subida
SetLongRAMAddress(CLKINTCR,0b00000010);
do
{
    i = GetLongRAMAddress(RFSTATE);
}
while((i&0xA0) != 0xA0);
}

```

También se muestra el diagrama de flujo de la interrupción al micro por parte del MRF24J40 cuando recibe un paquete al ser encuestado por el lector.



A continuación se muestra una parte del código de interrupción cuando es recibido un paquete proveniente del lector. Este paquete puede ser un comando de lectura o un comando de escritura, el programa está desarrollado de forma tal que no se puedan borrar los datos de la memoria si previamente no se emitió un comando de lectura.

```
#int_EXT
```

```
EXT_isr() // se debe llegar aquí cuando se recepciona un paquete
```

```
{
```

```
    auto int INTSrc;
```

```
    auto long RXFIFOp = RX_FIFO;
```

```
    INTSrc = GetShortRAMAddress(ISRSTS);
```

```
    if(INTSrc & 0x08) // solo entrare si la fuente de la interrupción
```

```
    {           // fue la recepción de un paquete
```

```
        FlashLedRojo(); // indica transmission correcta
```

```
        RXBuffer.PacketLength = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.FrameControl.LSB = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.FrameControl.MSB = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.SequenceNumber = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE0 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE1 = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE2 = GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE3 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE4 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE5 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE6 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstMACAddrBYTE7 =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstPANIDLowBYTE =  
GetLongRAMAddress(RXFIFOp++);
```

```
        RXBuffer.DestAddrInf.DstPANIDHighBYTE =  
GetLongRAMAddress(RXFIFOp++);
```

```
RXBuffer.SrcAddrInfo.MACAddrBYTE0 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE1 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE2 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE3 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE4 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE5 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE6 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.MACAddrBYTE7 = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.PANIDLowBYTE = GetLongRAMAddress(RXFIFOp++);
RXBuffer.SrcAddrInfo.PANIDHighBYTE = GetLongRAMAddress(RXFIFOp++);

RXBuffer.Command = GetLongRAMAddress(RXFIFOp);
SetShortRAMAddress(RXFLUSH,0x01); // limpiar el buffer de recepción y
//prepararlo para otra captura
if((RXBuffer.Command == ERASE_MEMORY) &&
    (LastCommand == READ_MEMORY))
{
    TXBuffer.Data.seg = 0;
    TXBuffer.Data.min = 0;
    TXBuffer.Data.hor = 0;
    TXBuffer.Data.days= 0;
    if(TransmitData())
    {
        FlashLedRojo(); // indica transmision correcta
    }
    else
    {
        LED_ROJO = 1;
        delay_ms(3000); // indica error en la transmision
        LED_ROJO = 0;
    }
    LastCommand = RXBuffer.Command;
}
else
if(RXBuffer.Command == READ_MEMORY)
```

```
    {  
        if(TransmitData())  
        {  
            FlashLedRojo(); // indica transmision correcta  
        }  
        else  
        {  
            LED_ROJO = 1;  
            delay_ms(3000); // indica error en la transmision  
            LED_ROJO = 0;  
        }  
        LastCommand = RXBuffer.Command;  
    }  
}  
}
```

### 3.3 Diseño y construcción del lector

Al igual que el sistema de adquisición de datos, el esquema eléctrico del lector (figura 3.3) se sustenta en un microprocesador PIC16LF73 y el Chip tranceptor MRF24J40.

Se alimenta de una batería de 9V y se utiliza un estabilizador de voltaje LM317 para obtener los 3.3V para la alimentación de ambos circuitos integrados.

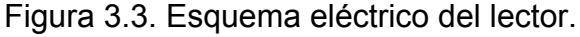
Con el objetivo de dar información al usuario sobre el estado de la batería, el LED rojo comenzará a parpadear una vez que la batería se descargue por debajo de 8V, indicando que debe ser remplazada. Para lograr esto se utiliza una referencia de voltaje de 1.2V conectada al terminal RA3, ya que éste es una de las entradas analógicas del microcontrolador. Al mismo tiempo, por otra de las entradas analógicas, en este caso RA0, se conecta un divisor de voltaje que estará alimentado directamente de la batería de 9V con el objetivo de que cuando el voltaje sea igual o menor que 8V este divisor dé a su salida un voltaje igual al de la referencia de voltaje, cuando la diferencia entre ambos sea igual a cero se activa el parpadeo del LED rojo indicando batería baja.



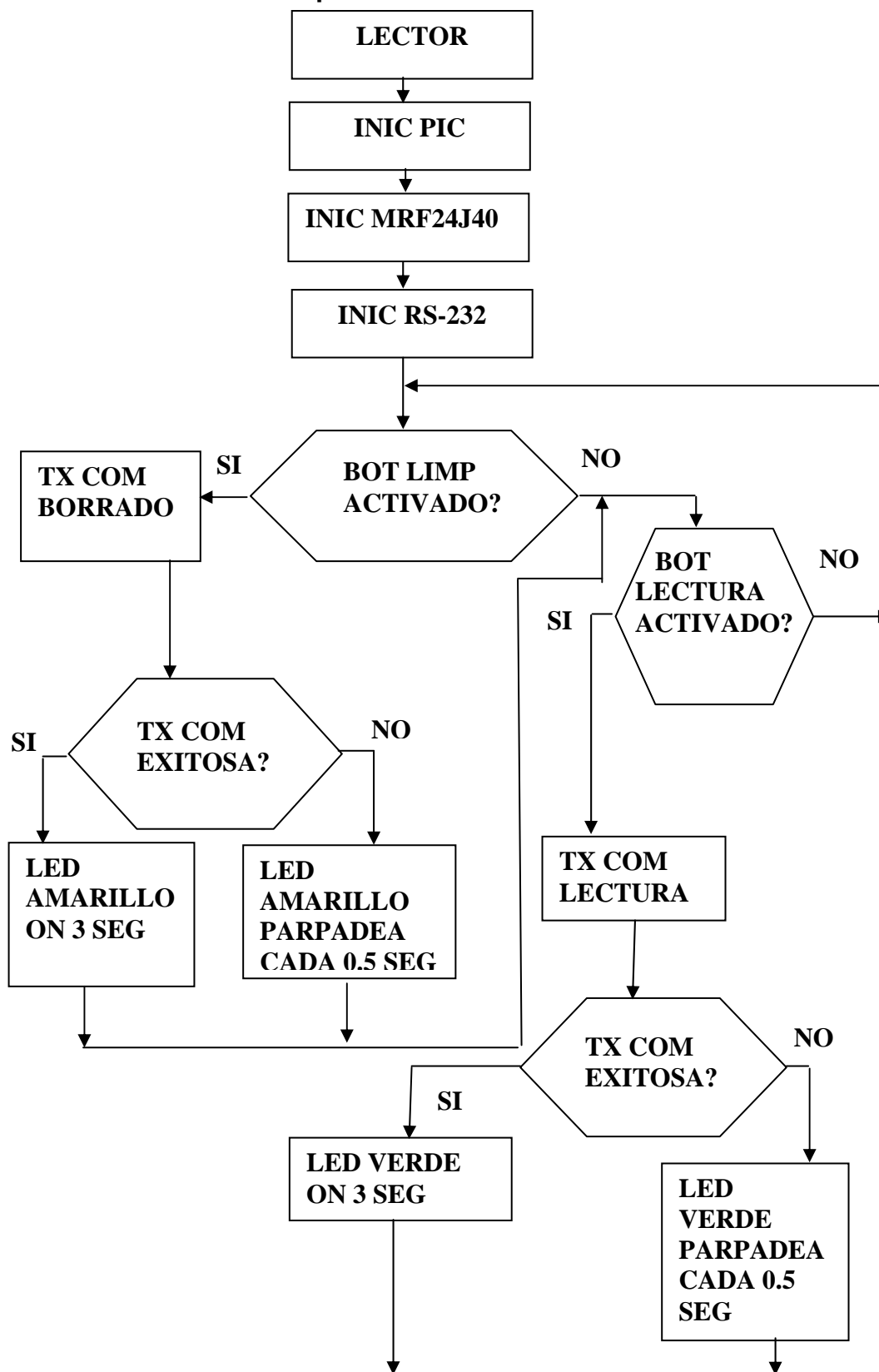
Otro aspecto a destacar en este esquema eléctrico es la alimentación del circuito integrado MAX232. Éste se alimenta cuando es despertado el microprocesador por alguna interrupción externa, ya sea proveniente del MRF24J40 o del botón de encendido, esto se logra activando un transistor PNP cuya base está conectada al terminal RC2 que pasa a un nivel bajo cuando el micro recibe cualquier interrupción externa, en el colector de este transistor se encuentran conectados la alimentación de la MAX232 y la referencia de voltaje AD1580 de 1.2V. La referencia de voltaje también se conecta a este nodo para evitar que consuma energía de forma innecesaria, cuando el lector está apagado.

La comunicación entre el PIC y el MRF24J40 fue analizada en la sección anterior y además se utilizan otras líneas de puertos para el enlace de ambos, como son: RC0 para el reset, RC1 para el WAKE, RB0 para la interrupción tanto del MRF24J40, como del botón ON-OFF y RA5 para el CS.

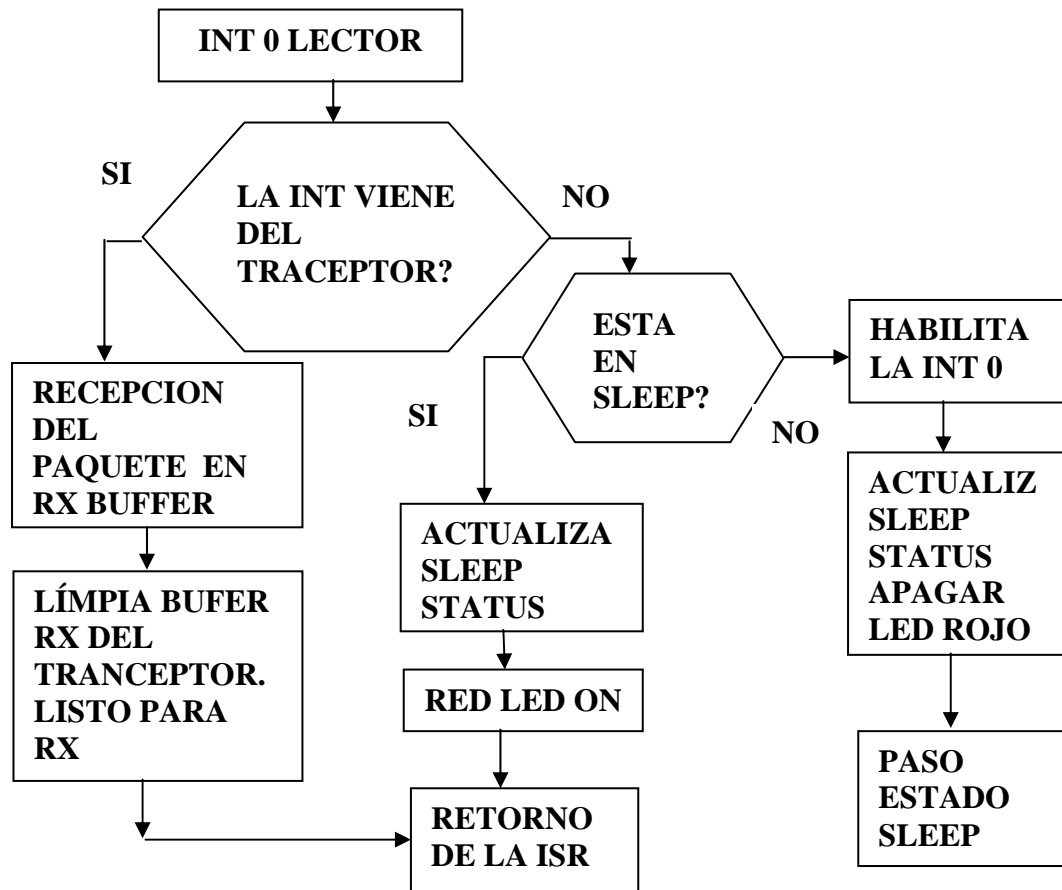
La comunicación con la PC para la descarga de los datos almacenados en la memoria del microcontrolador se realiza vía RS232, para lo cual está implementado el circuito MAX 232.



### 3.4. Soluciones de software para el lector



En la página anterior se muestra el diagrama de flujo del programa principal del lector y seguidamente el diagrama de la rutina de interrupción.



Para una mejor comprensión del diagrama de flujo mostrado arriba, a continuación se muestra un fragmento de código de interrupción cuando se recibe un paquete proveniente del contador, que en este caso son los datos del tiempo trabajado por el quemador de la Kalfrisa.

```

#int_EXT
EXT_isr()
{
    auto int INTsrc;
    auto long RXFIFO = RX_FIFO;
    INTsrc = GetShortRAMAddress(ISRSTS);
    FlashLedRojo();
    if(INTsrc & 0x08)

```

```
{
    // la interrupción fue provocada por recepción de un paquete
    RXBuffer.PacketLength = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.FrameControl.LSB = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.FrameControl.MSB = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SequenceNumber = GetLongRAMAddress(RXFIFOp++);

    RXBuffer.DestAddrInf.DstMACAddrBYTE0 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE1 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE2 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE3 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE4 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE5 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE6 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstMACAddrBYTE7 = GetLongRAMAddress(RXFIFOp++);

    RXBuffer.DestAddrInf.DstPANIDLowBYTE = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.DestAddrInf.DstPANIDHighBYTE = GetLongRAMAddress(RXFIFOp++);

    RXBuffer.SrcAddrInfo.MACAddrBYTE0 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE1 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE2 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE3 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE4 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE5 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE6 = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.MACAddrBYTE7 = GetLongRAMAddress(RXFIFOp++);

    RXBuffer.SrcAddrInfo.PANIDLowBYTE = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.SrcAddrInfo.PANIDHighBYTE = GetLongRAMAddress(RXFIFOp++);

    RXBuffer.Data.seg = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.Data.min = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.Data.hor = GetLongRAMAddress(RXFIFOp++);
    RXBuffer.Data.days = GetLongRAMAddress(RXFIFOp);

    SetShortRAMAddress(RXFLUSH,0x01); // limpiar el buffer de recepción y
                                     //prepararlo para otra captura
    //cont[7] = (char)(RXBuffer.Data.hor + 0x30); // conversion int-to-char manual
    //cont[9] = (char)(RXBuffer.Data.min + 0x30);
    //cont[11] = (char)(RXBuffer.Data.seg + 0x30);
    printf(cont[7],"%U",RXBuffer.Data.hor);
    printf(cont[9],"%U",RXBuffer.Data.min);
    printf(cont[11],"%U",RXBuffer.Data.seg);
    putstr(cont,sizeof(cont));
}
else
{
    // la interrupción fue provocada por la tecla ON/OFF
    if(SLEEPStatus == FALSO)
```

```
    {
        // Se ha ordenado un apagado
        enable_interrupts(INT_EXT);
        COMN_SERIE = 1;      // apagar el led ON
        SLEEPStatus = VERDADERO;
        sleep();
    }
    else
    {
        // estaba dormido y fue despertado
        SLEEPStatus = FALSO;
        COMN_SERIE = 0;    // prender el led rojo de encendido
    }
}
}
```

### 3.5 Herramientas de diseño utilizadas durante la elaboración del proyecto.

El diseño de las soluciones de software del sistema propuesto, basado en la aplicación de microcontroladores PIC, se realizó mediante la utilización de las siguientes herramientas de diseño destinadas a aplicaciones con este tipo de microcontroladores: el Ensamblador-compilador CCSC, la plataforma de desarrollo MPLAB IDE y sus potencialidades para la Simulación y programación del programa creado y el Depurador-programador ICD2.

1. El CCSC es un Software Compilador y Ensamblador, que permite crear, en lenguaje C, el código fuente del programa, compilarlo y convertirlo a código en lenguaje ensamblador, según el conjunto de instrucciones básicas del PIC seleccionado. El mismo consta de un programa de entrada con asistente descriptivo para definir en la aplicación a diseñar: la utilización de las entradas y/o salidas para cada una de las líneas de puerto del PIC, la frecuencia y tipo de cristal de cuarzo, el uso del Conversor A/D interno, de los Temporizadores, del perro guardián (WDT), del puerto serie SCI, de las interrupciones que se utilizarán, entre otras opciones.
2. El MPLAB-IDE es una Plataforma de Desarrollo Integrada bajo Windows, con múltiples prestaciones, que permite escribir el programa para los PIC en lenguaje ensamblador o en lenguaje C, crear proyectos, ensamblar o compilar,

simular el programa y finalmente programar el componente, si se cuenta con el programador adecuado.

3. El Depurador ICD2 es una herramienta que tiene dos finalidades: por un lado es un depurador en circuito que facilita la depuración y puesta a punto del programa de aplicación, haciendo que el dispositivo PIC pueda ejecutar dicho programa paso a paso, con puntos de ruptura, tiempo real, etc. Por otro lado el ICD2 incluye el hardware y firmware necesario para realizar las tareas propias de grabación. El mismo se controla desde el software MPLAB IDE y se puede conectar a un puerto serie o a un puerto USB de la PC.

En el presente trabajo de maestría se requiere, además, del uso de otras conocidas herramientas asistidas por PC para el diseño de los circuitos impresos de la aplicación (ORCAD) y para la simulación eléctrica y lógica de los circuitos (PROTEUS).

Una vez realizado el montaje de los componentes en el circuito impreso o protoboard y el ajuste del hardware, se procede a la emulación del programa con el Emulador ICD2, empleando el hardware del prototipo para el ajuste de cada uno de los módulos del programa y lograr la versión final del programa principal. El programa obtenido es grabado sobre el microprocesador PIC, usando el propio ICD2 y la herramienta de diseño PICDEM plus2 que incluye un socket de programación. Se prueban las prestaciones técnicas del equipo, para detectar errores y realizar los ajustes finales del Software hasta lograr una absoluta compatibilidad entre el programa desarrollado y las soluciones de hardware del prototipo.

### **3.6 Programa residente en la PC para la descarga de los datos del lector**

El Programa residente en la PC para establecer la comunicación por puerto RS-232 entre el lector y la PC fue elaborado con el Software LabWindows/CVI. Este programa realiza las siguientes acciones: inicialización y configuración del puerto RS-232 de la PC, establecimiento del protocolo de transmisión y recepción de los datos, el almacenamiento en un fichero de texto y la visualización de las lecturas de los datos almacenados en la memoria del PIC, además brinda la

posibilidad de limpiar la memoria del lector una vez que los datos hayan sido descargados a la PC.

Al ejecutar el programa se mostrará en pantalla la ventana de presentación del mismo, la cual se muestra en la figura 3.4.

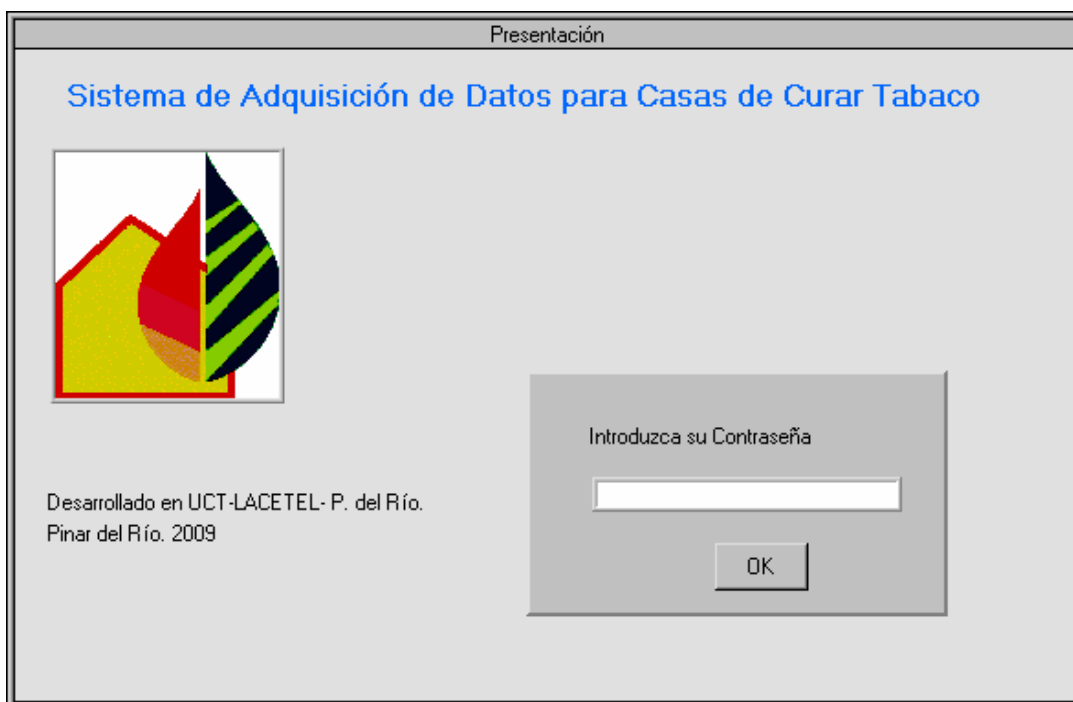


Figura 3.4. Ventana de presentación del programa.

Después de introducir la contraseña del usuario aparecerá el panel principal, que se muestra en la figura 3.5, el cual contiene una serie de botones de comando para la ejecución de las operaciones de trabajo.



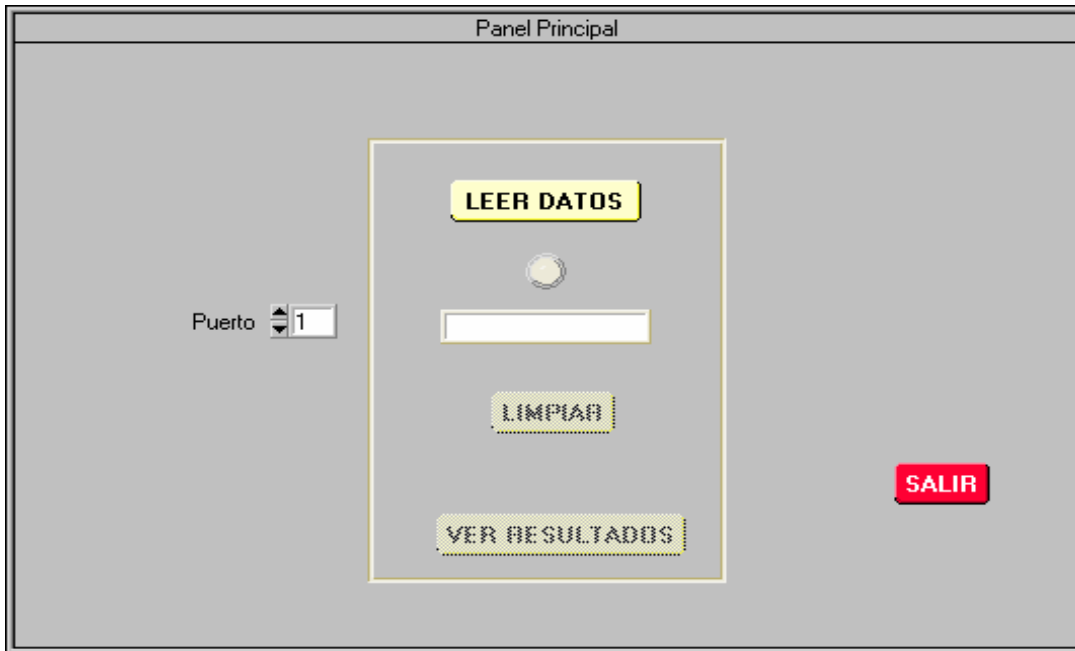


Figura 3.5. Panel principal del programa.

Para lograr la correcta comunicación entre el programa y el lector es necesario seleccionar el puerto que se encuentre desocupado en la PC. La configuración de éste se realiza mediante el panel que se muestra en la figura 3.6.

Una vez seleccionado el puerto se oprime el botón “LEER DATOS” para establecer la comunicación, si ésta por alguna razón no es exitosa aparece un mensaje avisando del error. Si por el contrario la comunicación fue exitosa se activa el botón “VER RESULTADOS” que permite al usuario visualizar en pantalla un documento de texto que contiene los datos leídos del lector. Este documento puede ser guardado en una carpeta en la PC de trabajo, para ser procesado posteriormente si se desea.

Después que los datos hayan sido descargados se activa el botón “LIMPIAR” dando la posibilidad de borrar la memoria del lector, lo que será indicado mediante el parpadeo del LED rojo.

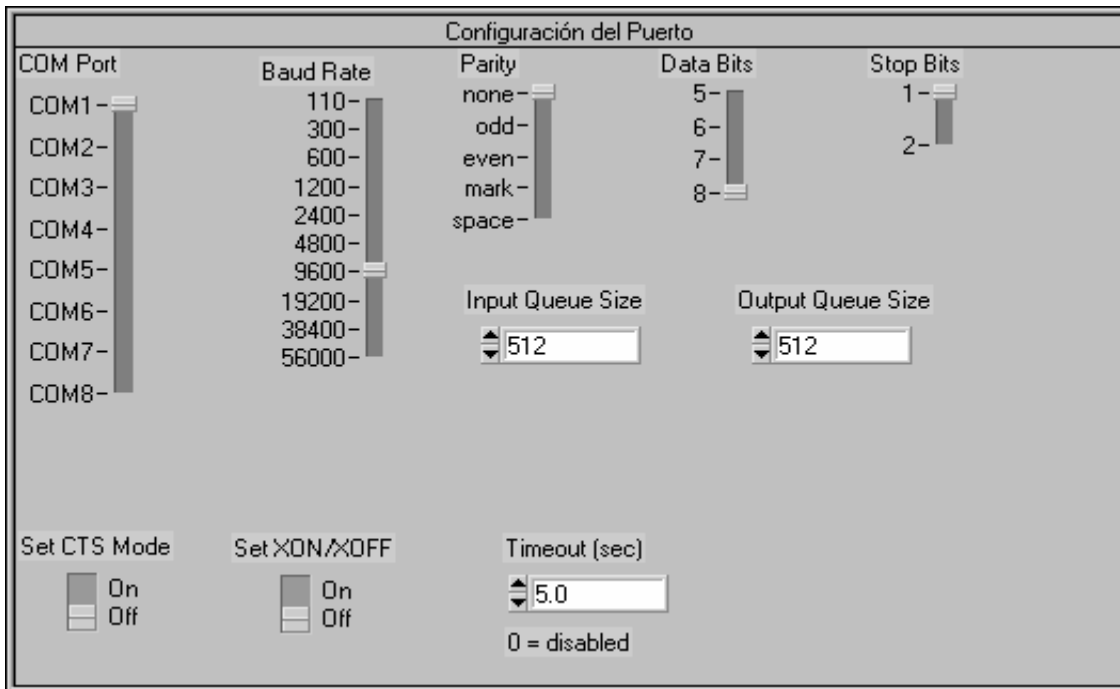


Figura 3.6. Panel de configuración del puerto.

A continuación se presenta un fragmento de la función leer datos del programa para ilustrar como es que los datos son leídos y transmitidos a la PC.

```
int CVICALLBACK Leer_datos (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int i,resp,status;
    switch (event)
    {
        case EVENT_COMMIT:
            Configurar();
            for(i=0;i<80;i++){
                datos[i]=datos_leidos[i]=0;
            }
            ComWrtByte (comport, 12);/*Enviar un 12 para lectura*/
            ComRd (comport, datos_leidos, 80);//leer 80 datos (bytes)

            RS232Error=ReturnRS232Err();
            if(RS232Error !=0){
                DisplayRS232Error ();//si es 0, OK y continúa
            }
            SetCtrlAttribute (panel_princHandle, PANEL_TEXTBOX, ATTR_VISIBLE, 0);
            SetCtrlVal (panel_princHandle, PANEL_LED, 0);
    }
}
```

---

```

SetCtrlAttribute(panel_princHandle, PANEL_COMMANDBUTTON_LIMPIAR,
ATTR_DIMMED, 1);
SetCtrlAttribute(panel_princHandle, PANEL_RESULT_BUTTON, ATTR_DIMMED, 1);
    }
    else{
        for(i=0;i<80;i++){
            datos[i]=datos_leidos[i];
            if(datos[i]<0)
                datos[i]=128+(255-(abs(~(datos[i]))+128));
            //cuando el dato es >127 (no cabe en tipo char),por lo que se obtiene negativo,éste
            //hay que convertirlo en tipo int y se recupera su valor inicial. la función ComRd(...)solo
            //admite datos tipo char)
        }
        GetProjectDir (dirname);
        status = FileSelectPopup (dirname, "nombre fichero.txt", "*.txt",
            "DataFile Storage", VAL_SAVE_BUTTON, 0, 0,
            0, 1, pathname);
        if (status != VAL_NO_FILE_SELECTED)
            resultados = fopen (pathname, "w+");

        fprintf(resultados,"\nInforme de Lecturas del tiempo de funcionamiento de cada
sad");
        fprintf(resultados,"\nFecha de lectura: %s Hora: %s \n", DateStr(),
TimeStr());
        fprintf(resultados,"-----\n");
        fprintf(resultados,"\nDispositivo Horas Minutos \n");

        for(i=0;i<77;i+=4){ //separar la cadena de datos de 4 en 4. Son 80 elementos,se
pone 77 porque la i se incrementa en 4
            cod_device=datos[i];
            horas=datos[i+1];
            minutos=datos[i+2];

            {
                default:
                    CopyString (mensaje, 0, mensaje5, 0, -1);
                    break;
            }

            fprintf(resultados,"\n %d %d %d %s
\n",cod_device,horas,minutos,mensaje);
        }
        fprintf(resultados,"\n\nControlado por: _____");
        fclose(resultados);
        SetCtrlAttribute (panel_princHandle, PANEL_TEXTBOX, ATTR_VISIBLE, 0);
        SetCtrlVal (panel_princHandle, PANEL_LED, 0);
        SetCtrlAttribute(panel_princHandle, PANEL_COMMANDBUTTON_LIMPIAR,
ATTR_DIMMED, 0);
        SetCtrlAttribute(panel_princHandle, PANEL_RESULT_BUTTON, ATTR_DIMMED, 0);

    }
    break;

```

```
        case EVENT_RIGHT_CLICK:
            MessagePopup ("Función",
                           "Establece la comunicación entre el
dispositivo y la PC\n"
                           "lee y guarda los datos en un fichero
de texto" );
            break;
        }
    return 0;
}
```

### 3.7. Gastos del proyecto

Para el análisis del gasto económico en que se incurrió en el proyecto se realizó una ficha de costo en la que se tuvo en cuenta el gasto de materiales empleados en el diseño, los gastos en salario de los participantes así como los gastos indirectos tales como electricidad consumida, papel y otros medios. En la tabla 12 se muestra el gasto de salario y la tabla 13 muestra la ficha de costo.

Tabla 12. Gasto de salario del proyecto

| DESAGREGACION DE LOS GASTOS DE SALARIO DURANTE LA INVESTIGACIÓN       |  |                       | MODELO -TIPO                   |                      |                  |
|---|--|-----------------------|--------------------------------|----------------------|------------------|
| Sistema de Adquisición de datos para Baterías de Casa de Curar Tabaco |  |                       |                                |                      |                  |
| DESCRIPCION DELAS OPERACIONES   | CANTIDAD DE TRABAJADORES/OPERACIÓN/ACTIVIDAD | CATEGORIA OCUPACIONAL | SALARIO/HORA/CATEGORIA Y GRUPO | NORMA TIEMPO (HORAS) | GASTO DE SALARIO |
| 1   | 2  | 3                     | 5                              | 6                    | 7(5X6)           |
| Búsqueda de información en internet                                   | 1  | Investigador          | 2,93809                        | 60                   | 176,285          |
| Estudio de la información.  | 1  | Investigador          | 2,93809                        | 80                   | 235,047          |
| Consulta a personal especializado                                     | 3  | Investigador          | 2,93809                        | 54                   | 158,656          |
| Realización del proyecto.   | 3  | Investigador          | 2,93809                        | 75                   | 220,356          |
| Montaje de la maqueta.  | 1  | Investigador          | 2,93809                        | 16                   | 47,0094          |
| Pruebas evaluativas   | 3  | Investigador          | 2,93809                        | 96                   | 282,056          |
| Programación de las maquetas  | 3  | Investigador          | 2,93809                        | 40                   | 117,523          |
| <b>TOTAL</b>  | <b>15</b>                                    |                       |                                | <b>421</b>           | <b>1236,93</b>   |
| Estimulación en divisas por trab.                                     |  |                       |                                |                      | 12               |
| Estimulación en divisas total   |  |                       |                                |                      | 397,586          |
| Vacaciones 9.09%  |  |                       |                                |                      | 111,324          |
| Seguridad Social 12.5%  |  |                       |                                |                      | 168,532          |
| Seguridad Corto Plazo 1.5%  |  |                       |                                |                      | 20,2239          |
|   |  |                       |                                |                      |                  |
| <b>GASTOS DE FUERZA DE TRABAJO</b>                                    |  |                       |                                |                      | <b>1934,60</b>   |

Tabla 13 Ficha de costo del proyecto

| FICHA PARA DETERMINAR EL PRECIO Y SU COMPONENTE EN PESOS CONVERTIBLES                    |  |  |                          |              |    |
|--|--|--|--------------------------|--------------|----|
| EMPRESA: Instituto de Investigación y Desarrollo LACETEL (UCT)                           |  |  | CODIGO: 161,0,06712      |              |    |
| ORGANISMO: MIC   |  |  |                          |              |    |
| Código:<br><br>Unidad de Medida:<br><br>Precio de venta total:<br><br>Componente en CUC: |  | % de utilización:<br><br>Producción periodo ant: |                          |              |    |
|  |  |  |                          |              |    |
|  |  | Uno<br>2010,53                                   | % margen utilidad:       |              | 12 |
|  |  |  | % sobre costo en divisas |              |    |
| CONCEPTOS DE GASTOS  |  | FILA   | Total Unitario           | De ello: CUC |    |
| MATERIAS PRIMAS Y MATERIALES   |  | 1  | 76,5576                  | 59,8850      |    |
| Materia Prima y Materiales fundamentales   |  | 1,1  | 76,2176                  | 59,5450      |    |
| Combustible y Lubricante   |  | 1,2  | 0,0000                   | 0,0000       |    |
| Energía electrica  |  | 1,3  | 0,3400                   | 0,3400       |    |
| Agua   |  | 1,4  | 0,0000                   | 0,0000       |    |
| SUB TOTAL (GASTO ELABORACIÓN)  |  | 2  | 1718,5584                | 432,8920     |    |
| OTROS GASTOS DIRECTOS  |  | 3  | 10,8123                  | 8,3172       |    |
| Depreciación   |  | 3,1  | 0,0000                   | 0,0000       |    |
| Arrendamiento de equipos   |  | 3,2  | 0,0000                   | 0,0000       |    |
| Ropa y Calzado (trabj. Directos)   |  | 3,3  | 18,5040                  | 8,3172       |    |
| GASTOS DE FUERZA DE TRABAJO  |  | 4  | 898,1142                 | 232,3190     |    |
| Salario  |  | 4,1  | 722,7702                 |              |    |
| Vacaciones   |  | 4,2  | 65,0493                  |              |    |
| Contribución a la Seguridad Social (12,5% )  |  | 4,3  | 98,4774                  |              |    |
| Impuesto Utilización Fuerza de Trabajo (25%)   |  | 4,4  | 0,0000                   |              |    |
| Seguridad Social Corto Plazo (1.5%)  |  | 4,5  | 11,8173                  |              |    |
| Estimulación   |  | 4.6  | 0,0000                   | 232,3190     |    |
| GASTOS INDIRECTOS DE DESARROLLO  |  | 5  | 386,2637                 | 60,4134      |    |
| Depreciación   |  | 5.1  | 17,8654                  | 14,3044      |    |
| Mantenimiento y reparación   |  | 5.2  | 0,0000                   | 0,0000       |    |
| GASTOS GENERALES Y DE ADMINISTRACION   |  | 6  | 517,1935                 | 168,3652     |    |
| Combustible y Lubricante   |  | 6.1  | 43,6861                  | 43,6861      |    |
| Energía electrica  |  | 6.2  | 11,7617                  | 11,7617      |    |
| Depreciación   |  | 6.3  | 17,8654                  | 14,3044      |    |

|   |  |     |                 |               |
|---|--|-----|-----------------|---------------|
| Ropa y Calzado                            |  | 6.4 | 0,0000          | 1,7642        |
| Alimentos                                 |  | 6.5 | 0,0000          | 22,5824       |
| Otros                                     |  | 6.6 | 443,8804        | 22,5824       |
| GASTOS DIST. Y VENTA                      |  | 7   | 0,0000          | 0,0000        |
| Combustible y Lubricante                  |  | 7.1 | 0,0000          | 0,0000        |
| Energía eléctrica                         |  | 7.2 | 0,0000          | 0,0000        |
| Depreciación                              |  | 7.3 | 0,0000          | 0,0000        |
| Ropa y Calzado                            |  | 7.4 | 0,0000          | 0,0000        |
| Otros                                     |  | 7.5 | 0,0000          | 0,0000        |
| GASTOS BANCARIOS                          |  | 8   | 6,1746          | 3,4772        |
| GASTOS TOTALES                            |  | 9   | 1795,1160       | 532,7770      |
| Margen utilidad s/ base autorizada        |  | 10  | <b>215,4139</b> |               |
| <b>PRECIO (Según los establecido MFP)</b> |  | 11  | <b>2010,53</b>  |               |
| <b>COMPONENTE EN PESOS CONVERTIBLES</b>   |  | 13  |                 | <b>532,78</b> |

En esta tabla se puede apreciar que el precio de desarrollo de este producto asciende a 210,53 pesos en moneda total, de ellos:

532,18 en moneda libremente convertible (CUC) y 2510,60 en CUP.

## *Conclusiones.*

El Trabajo de Diploma ha cumplido satisfactoriamente todos los objetivos trazados al inicio de la investigación y además cumple las especificaciones de calidad requeridas por el cliente para la aplicación de este sistema a la cura del tabaco como un medio para llevar las estadísticas necesarias para hacer este proceso cada día más eficiente.

En la investigación se alcanzaron los siguientes resultados:

1. Se obtuvo un hardware que responde a las necesidades planteadas para la obtención de los datos del proceso de curado del tabaco sin que ninguna persona que no sea autorizada pueda tener acceso a los datos registrados por el SAD, lo que permite evaluar el funcionamiento correcto de las soluciones obtenidas.
2. Se realizó un estudio profundo del protocolo de comunicación perteneciente a la norma IEEE 802.15.4 para su aplicación en el diseño, utilizando el chip de comunicación MRF24J40, con lo que se logró la comunicación entre éste y el PIC por medio de la interfaz de comunicación serie (SPI) y la comunicación inalámbrica entre el Sistema de Adquisición de Datos y el lector.
3. Las soluciones tanto de hardware como de software alcanzadas en este proyecto sirven de punto de partida para la realización de otros proyectos en la esfera de la automatización y control a distancia de procesos tecnológicos por complicados que estos fueran.
4. Se desarrolló un programa CVI residente en PC para la descarga de los datos del lector, los que pueden ser salvados en una carpeta y usados como estadística para la valoración de la eficiencia del proceso de curado del tabaco en una batería de Kalfrisas



Valorando el cumplimiento satisfactorio de los objetivos propuestos al comienzo de esta investigación, y considerando la novedad, la actualidad del tema y el elevado rigor científico-técnico de las soluciones alcanzadas, se llega a la conclusión, de que los resultados obtenidos superaron las expectativas iniciales, y que se cuenta con los elementos tecnológicos necesarios, para aplicar este tipo de comunicación al diseño de complejos sistemas automatizados para el control de distintos parámetros tecnológicos.

## *Bibliografía*

1. Copyright © 2000 Microchip Technology Inc. PIC16F73/16F74/16F76/16F77. Data Sheet. USA, 2000.
2. Copyright © 2006 ZigBee Standards Organization. ZigBee APL Layer Protocol Implementation Conformance (PICS) proforma. ZigBee Alliance, Inc. 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, December 1, 2006.
3. Copyright © 2006 ZigBee Standards Organization. ZigBee NWK Layer protocol implementation conformance (PICS) proforma. ZigBee Alliance, Inc. 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, December 1, 2006. Copyright © 2006 Microchip Technology Inc.
4. Copyright © 2006 ZigBee Standards Organization. ZigBee IEEE 802.15.4 PHY & MAC Layer Test Specification. ZigBee Alliance, Inc. 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, December 1, 2006.
5. Copyright © 2006 Microchip Technology Inc MRF24J40 Data Sheet IEEE 802.15.4™ 2.4 GHz RF Transceiver. USA, 2006.
6. Manual de Programación de las Herramientas de Diseño CCS. Software Ensamblador de lenguaje C al lenguaje nemotecnico del PIC familia 16XX. CD Rom Microchip. 2002.
7. Manual de Programación de las Herramientas de Diseño MPLAB. Software de Simulación y Debug del programa para aplicaciones con PIC. CD Rom Microchip 2002.
8. Ramírez Torres Frank Ernesto. Tesis de grado (Soluciones técnicas para la comunicación inalámbrica con el protocolo ZigBee empleando los microcontroladores PIC).
9. Álvarez López Ihosvany: Aportaciones al diseño e implementación de controladores difusos: aplicados al curado de tabaco en hojas. Tesis doctoral
10. <http://www.ieee802.org/15/index.html>.
11. <http://www.awflowmeters.com>
12. <http://www.Marvitech.com>

## *Referencias bibliográficas*

- [1] Álvarez López Ihosvany: Aportaciones al diseño e implementación de controladores difusos: aplicados al curado de tabaco en hojas. Tesis Doctoral
- [2] Copyright © 2000 Microchip Technology Inc.  
PIC16F73/16F74/16F76/16F77. Data Sheet. USA, 2000.
- [3] Copyright © 2006 Microchip Technology Inc MRF24J40 Data Sheet IEEE 802.15.4™ 2.4 GHz RF Transceiver. USA, 2006.
- [4] Copyright © 2006 ZigBee Standards Organization. ZigBee IEEE 802.15.4 PHY & MAC Layer Test Specification. ZigBee Alliance, Inc. 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, December 1, 2006.
- [5] Manual de Programación de las Herramientas de Diseño CCS. Software Ensamblador de lenguaje C al lenguaje nemotecnico del PIC familia 16XX. CD Rom Microchip. 2002.
- [6] Manual de Programación de las Herramientas de Diseño MPLAB. Software de Simulación y Debug del programa para aplicaciones con PIC. CD Rom Microchip 2002.